

PCTWORLD INTELLECTUAL PROPERTY ORGANIZATION
International Bureau

INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

| | | |
|--|-----------|--|
| (51) International Patent Classification ⁷ : G06K 1/12 | A1 | (11) International Publication Number: WO 00/023154 (43) International Publication Date: 5 October 2000 |
| (21) International Application Number: PCT/US00/08405 (22) International Filing Date: 30 March 2000 (30.03.00) (30) Priority Data: 60/126,999 30 March 1999 (30.03.99) US (71) Applicant: INTERMEC IP CORP. [US/US]; 21900 Burbank Boulevard, Woodland Hills, CA 91367-7418 (US). (72) Inventors: MENKEN, Jeffrey; 15433 Country Club Drive #G106, Mill Creek, WA 98012 (US). WIKLOF, Christopher, A.; 3531 - 99th Street S.E., Everett, WA 98208 (US). SAMUEL, Robert, A.; 1702 Orca View Lane, Everett, WA 98203 (US). (74) Agents: ABRAMONTE, Frank et al.; Perkins Coie LLP, 1201 Third Avenue, Suite 4800, Seattle, WA 98101-3099 (US). | | (81) Designated States: JP, European patent (AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE). Published <i>With international search report.</i> |
| (54) Title: A METHOD AND SYSTEM FOR AUTOMATIC DATA COLLECTION ("ADC") SYMBOL PREPARATION (57) Abstract A method and system for generating automatic data collection ("ADC") symbols and controlling the production and printing of the ADC symbols on a symbol production device or a symbol printing device. An ActiveX control may be provided to a computing application so as to enable the application to produce and print bar code labels. A bitmap image containing instructions for printing bar-code labels may be sent from the application to a printer driver via a general purpose graphical display interface provided by an operating system. The general purpose graphical display interface has no requirement for understanding the bitmap image and passes the bitmap image through to the printer driver that understands the bar-code instructions in the bitmap image. The invention is applicable for the production of other ADC symbols utilizing ActiveX technology. The invention is also applicable to any operating systems. The invention may also utilize technologies other than ActiveX, such as Java. | | |

Best Available Copy

FOR THE PURPOSES OF INFORMATION ONLY

Codes used to identify States party to the PCT on the front pages of pamphlets publishing international applications under the PCT.

| | | | | | | | |
|----|--------------------------|----|--|----|--|----|--------------------------|
| AL | Albania | ES | Spain | LS | Lesotho | SI | Slovenia |
| AM | Armenia | FI | Finland | LT | Lithuania | SK | Slovakia |
| AT | Austria | FR | France | LU | Luxembourg | SN | Senegal |
| AU | Australia | GA | Gabon | LV | Latvia | SZ | Swaziland |
| AZ | Azerbaijan | GB | United Kingdom | MC | Monaco | TD | Chad |
| BA | Bosnia and Herzegovina | GE | Georgia | MD | Republic of Moldova | TG | Togo |
| BB | Barbados | GH | Ghana | MG | Madagascar | TJ | Tajikistan |
| BE | Belgium | GN | Guinea | MK | The former Yugoslav Republic of Macedonia | TM | Turkmenistan |
| BF | Burkina Faso | GR | Greece | ML | Mali | TR | Turkey |
| BG | Bulgaria | HU | Hungary | MN | Mongolia | TT | Trinidad and Tobago |
| BJ | Benin | IE | Ireland | MR | Mauritania | UA | Ukraine |
| BR | Brazil | IL | Israel | MW | Malawi | UG | Uganda |
| BY | Belarus | IS | Iceland | MX | Mexico | US | United States of America |
| CA | Canada | IT | Italy | NE | Niger | UZ | Uzbekistan |
| CF | Central African Republic | JP | Japan | NL | Netherlands | VN | Viet Nam |
| CG | Congo | KE | Kenya | NO | Norway | YU | Yugoslavia |
| CH | Switzerland | KG | Kyrgyzstan | NZ | New Zealand | ZW | Zimbabwe |
| CI | Côte d'Ivoire | KP | Democratic People's Republic of Korea | PL | Poland | | |
| CM | Cameroon | KR | Republic of Korea | PT | Portugal | | |
| CN | China | KZ | Kazakhstan | RO | Romania | | |
| CU | Cuba | LC | Saint Lucia | RU | Russian Federation | | |
| CZ | Czech Republic | LJ | Liechtenstein | SD | Sudan | | |
| DE | Germany | LK | Sri Lanka | SE | Sweden | | |
| DK | Denmark | LR | Liberia | SG | Singapore | | |
| EE | Estonia | | | | | | |

A METHOD AND SYSTEM FOR AUTOMATIC DATA COLLECTION ("ADC") SYMBOL PREPARATION

TECHNICAL FIELD

The present invention is directed to the field of preparing automatic
5 data collection ("ADC") symbols.

BACKGROUND OF THE INVENTION

Automatic Data Collection ("ADC") device platforms, such as
ADC device platforms equipped with bar-code readers, have received increasing
commercial attention in the past few years. ADC device platforms, such as hand-
10 held data collection terminals, or hand-held personal computers, have been
widely implemented in the retail marketplace and have garnered increasing use
in a diverse range of applications. The ever-decreasing cost and size of ADC
device platforms has facilitated their entry into a wide variety of commercial,
institutional, and governmental settings.

15 An ADC device platform having a bar-code reader adeptly accesses
and retrieves data stored in the form of a bar-code label. Data representing
virtually any product or service found in the stream of commerce can be encoded
in a bar-code label for later access by an ADC device platform having a bar-code
reader. Bar-code readers include laser scanners as well as other means of
20 collecting product information, such as a bar-code wand, a still camera or an area
imager. As is well known, other ADC data formats exist, including Radio
Frequency ("RF") tags, resonators, SmartCards, and magnetic strips.

In a conventional ADC environment, specialized tools prepare the
ADC symbols, such as bar codes, that may be affixed to, or otherwise associated
25 with, a product or service. Despite the ever-increasing robustness found in ADC
device platforms, ADC symbols must still be produced with a degree of precision
for an ADC device to retrieve the data encoded in an ADC symbol. A

specialized computer programming application represents one tool for preparing ADC symbols.

Figure 1 is a high-level block diagram of a general-purpose computer system 100 upon which a specialized bar-code label application 131 executes. The computer system 100 may be contained within a facility for producing the bar-code labels. The computer system 100 typically contains a central processing unit ("CPU") 110, input/output devices 120, and a computer memory ("memory") 130. A computer-readable media drive 121, such as a CD-ROM, is conventionally included among the input/output devices 120, as well as a printer 122, a storage device 123, and a connection 124. A hard disk drive represents a typical storage device 123. The drive 121 may be used to install software products, such as the label application 131. The connection 124 may connect the computer system 100 to servers and transaction sources from which transactions are received. For example, the connection 124 may be used to connect the computer system 100 to one or more computer telephony interfaces, World Wide Web servers, or World Wide Web browsers.

The memory 130 typically contains the label application 131 and a printer driver 132 for the printer 122. The printer driver 132 contains instructions for controlling the operation of the printer 122.

A bar-code label maker engages the functionality provided by the label application 131 to produce bar-code labels. The label application 131 contains all the instructions necessary to produce bar-code labels. The label maker may be a human worker or another computer program. Once the label maker has produced at least one bar-code label, the label maker may instruct the label application 131 to print to the bar-code labels. The label application 131 provides the necessary instructions to the printer driver 132 which then converts the instructions into appropriate commands for directing the printer 122 to print the bar-code labels.

The label application 131 is a specialized computer program that does not readily share instructions for producing bar-code labels with other

computer programs. The specialized nature of the label application 131 frequently renders the label application 131 less familiar, intuitive, and user-friendly than more commonly used computer programs. Moreover, imbuing the label application 131 with the comfortable, user-accessible style of more familiar
5 computer programs requires constant attention by the program's developers to keep up with changing user interface styles. In addition, maintaining of the label application 131 requires costly programmer attention to every embodiment of the application, attention to details far beyond merely preparing bar-code labels. Consequently, developers of bar-code label applications must either continuously
10 expend costly programmer resources to maintain their bar-code label applications or settle for less expensive, but inflexible and frequently obsolete applications.

BRIEF DESCRIPTION OF THE DRAWINGS

Note that similar elements and steps in the figures have the same reference number. For ease in identifying the discussion of any particular
15 element, the most significant digit in a reference number typically refers to the figure number in which that element is first introduced (*e.g.*, element*204 is first introduced and discussed with respect to Figure 2).

Figure 1 is a high-level block diagram of a sample general-purpose computer system 100 upon which a conventional bar-code label application 131
20 executes.

Figure 2 illustrates a bar-code ActiveX control 202 that provides functionality for producing bar-code labels embedded into a general-purpose WINDOWS® computing application 201.

Figure 3 illustrates an ActiveX control and some of its interfaces,
25 such as the bar-code ActiveX control 202 shown in Figure 2.

Figure 4 is a diagram of a sample computing system 430 that produces bar-code labels using the application 201 and the bar-code ActiveX control 202.

Figure 5 illustrates embodiments of a bar-code label template 405.

Figure 6 illustrates a template properties page 601 that can be associated with the bar-code template 405.

Figure 7 illustrates a bar-code properties table 701 that can be associated with the bar-code template 405.

5 Figure 8 illustrates a human-readable properties table 801 that may be associated with the bar-code template 405.

Figure 9 illustrates a font selection table 901 that may be associated with the bar-code template 405.

10 Figure 10 illustrates a general properties table 1001 that may be associated with the bar-code template 405.

Figure 11 illustrates a serialization property table 1101 that may be associated with the bar-code template 405.

Figure 12 illustrates a bitmap image 1201, according to an embodiment of the invention.

15 Figure 13A illustrates the bitmap image 1201 printed as a bar-code label 1301 on a non-proprietary printer, or with a non-proprietary printer driver, with the security bits enabled, according to an embodiment of the invention.

Figure 13B illustrates the bitmap image 1201 printed as a bar-code label 1302 on a proprietary printer with the security bits enabled, or as printed on
20 a non-proprietary printer with the security bits disabled, according to an embodiment of the invention.

Figure 14 is a flowchart illustrating the operations of the bar-code ActiveX control 202, according to an embodiment of the invention.

Figure 15 illustrates the application 201 and the ActiveX control
25 202 operating in combination with an automation program 1501, according to an embodiment of the invention.

Figure 16 illustrates a bar-code ActiveX control 1604 utilized in a client/server environment, according to an embodiment of the invention.

Figure 17 illustrates a system for producing SmartCards, according
30 to an embodiment of the invention.

DETAILED DESCRIPTION OF THE ILLUSTRATED EMBODIMENTS OF THE INVENTION

Embodiments of the invention provide a method and system for passing data through an intermediary programming element in an acceptable format so that the intermediary programming element may forward the data to at least another programming element that may interpret the data. For example, one embodiment of the invention provides a method and system for generating automatic data collection ("ADC") symbols and controlling the production and printing of the ADC symbols on an ADC symbol production device, such as an ADC symbol printing device.

An ActiveX control may be provided to a computing application so as to enable the application to produce and print bar-code labels, according to an embodiment of the invention. A bitmap image containing instructions for printing bar-code labels may be sent from the application to a printer driver via a general purpose graphical display interface provided by an operating system, according to an embodiment of the invention. The general purpose graphical display interface has no requirement for understanding the bitmap image and passes the bitmap image through to the printer driver which understands the bar-code instructions in the bitmap image.

Embodiments of the invention are applicable to the production of other ADC symbols. For example, the invention may be utilized to provide the programming for SmartCards, the encoding of radio frequency ("RF") tags, the encoding for dipole device symbols, and the encoding for an ADC symbol suitable for acquisition by an optical character recognition ("OCR") device.

In one embodiment of the invention, the general purpose graphical display interface may be the graphical display interface ("GDI") provided by the WINDOWS® operating system. In another embodiment of the invention, the general purpose graphical display interface may be a graphical display interface provided by the Macintosh operating system ("Mac OS"). In yet other embodiments of the invention, the general purpose graphical display interface

may be a graphical display interface provided by other operating systems and other programming elements in a computing system.

Embodiments of the invention provide a method and system for producing ADC symbols in a local computing system. Embodiments of the invention provide a method and system for producing ADC symbols in a
5 networked computing system. Embodiments of the invention provide a method and system that may be utilized by both human users and robotic users in the preparation of ADC symbols. Embodiments of the invention provide a method and system suitable for use with technologies that automate the production of
10 ADC symbols. Embodiments of the invention further provide a help system and an instructional system ("a wizard") to support a user in the production of ADC symbols.

The method and system for preparing and manufacturing automatic data collection ("ADC") symbols may be utilized in preparing bar-code labels,
15 radio frequency ("RF") tags, resonators, SmartCards, magnetic strips, optical character recognition ("OCR") symbols, recorded speech symbols, two-dimensional ("2D") symbols, dipole devices (such as those recited in U.S. Patent No. 5,581,257), and any symbol having encoded data therein. The invention may be used to manufacture ADC symbols on any production device, such as a
20 printer, using any computing system having any operating system.

To more fully describe embodiments of the invention, their operation under a specific example is discussed. This example conveniently discusses embodiments of the invention, and therefore not necessarily represents all applications of the invention. Indeed, one skilled in the relevant art will
25 recognize that the invention can be practiced without one or more of the specific details, or with other symbols, methods, etc. In other instances, well-known structures or operations are not shown or not described in detail to avoid obscuring embodiments of the invention.

An embodiment of the invention provides a facility for preparing
30 bar-code labels by adding bar-code preparation functionality to an existing

program, such as the executable code for an existing program. A further embodiment of the invention provides a method for passing bar-code label printing instructions through an operating system's internal functions (that are bar-code unaware) to a printer driver that understands bar-code format specifications. A still further embodiment of the invention provides bar-code label preparation in a WINDOWS® operating system on an IBM-compatible personal computer ("PC"). WINDOWS® is an operating system produced by the Microsoft Corporation that provides a multitasking graphical user interface ("GUI") environment that runs on both MS-DOS-based computers and as a self-contained operating system. WINDOWS® provides a standard interface based on drop-down menus, windowed regions on a display screen, and a pointing device such as a mouse.

A bar-code object that provides user-selectable functions for producing bar-code labels may be embedded into any general-purpose WINDOWS® computing application, such as a general-purpose computing application 201 shown in Figure 2. The application 201 may be a database program, a word-processing program, or a spreadsheet program, for example.

The application 201 attains the ability to receive and process bar-code objects by adding a bar-code ActiveX control 202. ActiveX controls are reusable software components that may be used to add specialized functionality, such as animation and pop-up menus, to existing programs. ActiveX controls may be written in a variety of programming languages, including C, C++, Visual Basic, and Java. ActiveX controls are discussed further below.

Once a bar-code label has been prepared, the application 201, operating with the bar-code ActiveX control 202, sends instructions for printing the bar-code label to a graphical device interface ("GDI") 204. The GDI 204 provides specialized display processing functionality for the WINDOWS® operation system in the WINDOWS® environment.

WINDOWS® has been designed so that programs operating in WINDOWS®, such as the application 201, will function without difficulty using

any output device that WINDOWS® supports. Because many different display devices may be attached to an IBM PC and its compatibles, the WINDOWS' GDI 204 consolidates support of device-independent graphics on output devices such as video displays, printers, and plotters. Accordingly, the GDI 204 insulates
5 programs, such as the application 201, from the particular requirements and characteristics of different output devices. The GDI 204 also provides strong support of pixel-level manipulation.

Through the GDI 204, WINDOWS® "virtualizes" display hardware so that any program, such as the application 201, will operate with any
10 video board or any printer. Thus, programs need not directly access display device hardware, such as that of a display screen and/or a printer, but instead access the GDI 204 that provides for the display and printing of graphics and formatted text. Accordingly, since the GDI 204 provides display device interoperability, programs, such as the application 201, do not need to contain
15 functionality for determining display device characteristics.

The GDI 204 allows applications, such as the application 201, to draw on screens, printers, and other output devices. The GDI 204 is used by applications to display or print bitmapped text (including TrueType fonts), images, or other graphical elements. A bitmap is a data structure that represents
20 information in the form of a collection of individual bits. A bitmap may represent a bit image, such as the general shape of a font character or a bar-code element. The GDI 204 also draws dialog boxes, buttons, and other elements in a consistent style on display devices by calling the appropriate drivers and passing them appropriate information.

25 The GDI 204 calls routines in various driver files, such as the drivers for controlling printers and plotters. Different printers require different driver files. The GDI 204 allows WINDOWS®, rather than a program itself, to determine from the printer driver what tasks the printer driver can process and what tasks require assistance. For example, if the video hardware includes a
30 graphics coprocessor that can draw ellipses, then the GDI 204 can take advantage

of this hardware; otherwise, the GDI 204 calculates the points of the ellipse and passes the points to the driver.

When the GDI 204 works with output devices such as printers, the GDI 204 calls the appropriate printer driver and moves the image or document
5 directly to the printer, rather than re-formatting the image or document in a specialized printing language, such as PostScript. The GDI 204 provides hundreds of convenient functions for tasks such as drawing lines, circles, and polygons; rendering fonts, and querying devices for their output capabilities. Accordingly, the GDI 204 contains specialized areas for the production of
10 bitmaps 207, vector graphics 206, and fonts 205.

The GDI 204 does not support the drawing of bar codes. Like most ADC symbols, bar codes encode data in complicated formats for which the GDI 204 has no associated functionality. For example, even the GDI's bitmaps 207 are too primitive to process bar codes adequately.

15 Nevertheless, the bar-code ActiveX control 202 and a printer driver 209 may use a bitmap as a "Trojan horse" for passing bar-code data through the GDI 204. Once the application 201 and the bar-code ActiveX control 202 have prepared a bar-code label, instructions for printing the bar-code label may be transmitted to the printer driver 209 in the form of a bitmap suitable for
20 processing by the GDI's bitmaps 207.

The bitmaps 207 do not need to know that the information provided by the application 201 actually pertains to bar-code labels (or even an ADC symbol). The bitmaps 207 in conjunction with the GDI 204 determines the printer selected for the application 201, locates the appropriate printer driver
25 (e.g., the printer driver 209), and sends the bitmap image (containing the bar-code label instructions) to the printer driver 209. The printer driver 209 receives the bitmap image processed by the bitmaps 207 and examines the received bitmap image to locate the bar-code label instructions hidden within the bitmap image. The printer driver 209 then constructs a proper bar-code label suitable for

printing on a printer 211 and sends printing control instructions to the printer 211 to print the bar-code label.

Figure 3 illustrates an ActiveX control, such as the bar-code ActiveX control 202 shown in Figure 2. ActiveX is a set of technologies that enables software components to interact with one another, regardless of the language in which the components were created. ActiveX, developed by Microsoft, builds on Microsoft's well-known component object model ("COM"). COM provides a specification for building software elements that can be assembled into programs or add functionality to existing programs running on Microsoft WINDOWS® platforms. COM further allows different applications to communicate with each other.

ActiveX controls provide programming components that may be inserted in an existing program, such as an application 301, to provide functionality not otherwise available from the program, such as animation sequences, credit-card transactions, real-time video sequences, spreadsheet functions, or bar-code preparation functions. The application 301 may utilize an ActiveX control, such as a bar-code ActiveX control 302, without necessity for being re-compiled or re-linked.

The functionality provided by an ActiveX control, such as the bar-code ActiveX control 302, comprises four embodiments, each of which may be implemented by a group of interfaces 303-306. The bar-code ActiveX control 302 may provide the user interface 306 for its user interface functionality into an existing user interface 307, such as the user interface associated with the application 301. The bar-code ActiveX control 302 allows a "container" (e.g., the application 301) to invoke the ActiveX methods through the interface 303. The bar-code ActiveX control 302 sends events to the container (the application 301) through the interface 304. Events refer to a sequence of actions that occur in response to given actions by a user, a program, or an operating system. For example, a user may interact with the bar-code user interface 306 (through the user interface 307) in such a manner so as to trigger events processed through the

interface 304. The bar-code ActiveX control 302 may examine and modify properties associated with the container through the interface 305.

Figure 4 illustrates a sample computing system 430 for producing bar-code labels using the application 201 and the bar-code ActiveX control 202, according to an embodiment of the invention. The computing system 430 includes a monitor 401, a computer 408, a keyboard 414, a mouse 415, and the printer 211. The monitor 401 displays a graphical user interface ("GUI") 402 produced by the application 201. The GUI 402 represents a graphical portion of the user interface 307 shown in Figure 3 and includes pull-down menus 403 and a task bar 406. The computer 408 includes a central processing unit ("CPU") 409 and a memory 410. The memory 410 contains a copy of the application 201 and the bar-code ActiveX control 202.

The computing system 430 may also include a network interface 427a, a magnetic disk drive interface 428a, and an optical drive interface 429a, which may be respectively configured to operate in conjunction with a network 427, a magnetic disk 428, and an optical disk 429. For example, the application 201 and/or the bar-code ActiveX control 202 may be initially provided to the computing system 430 via the network 427, the magnetic disk 428, and/or the optical disk 429.

As previously discussed, the application 201 may be any type of computer program. For purposes of illustration, the application 201 shown in Figure 4 is a word processing application having text areas 404a and 404b. A user may send instructions to the application 201 by interacting with the keyboard 414 and the mouse 415.

The application 201 may receive an instruction for inserting a bar-code label template 405 between the text areas 404a and 404b. Accordingly, the application 201 directs the GUI 402 to display the label template 405 between the text areas 404a and 404b.

The application 201 may receive additional instructions regarding the label template 405. For example, the application 201 may receive

instructions regarding the inclusion of specific data in the label template 405, a bar-code symbology type, as well as other pertinent characteristics.

The application 201 may also receive instructions directing the printing of the label template 405. As discussed in Figure 2, the application 201 sends the GDI 204 a bitmap that will be provided to the printer driver 209. The printer driver 209 then instructs the printer 211 to print bar-code labels, such as bar-code labels 413a-413c. In a sample environment, the printer 211 prints the bar-code labels 413a-413c on a paper media 412 having self adhesive bar-code labels 413a-413c that may be removed easily from a non-stick liner forming part of the paper 412. Of course, other print media may be used, such as liner-less paper.

Figure 5 illustrates embodiments of the label template 405, which includes a bar-code label 501 and a code section 502 representing instructions to a printer driver, such as the printer driver 209. The code section 502 need not necessarily be shown on the GUI 402. In an initial display on the GUI 402, the bar-code label 501 may have a default value subject to change according to instructions received by the application 201.

The label template 405 may be acted upon when a mouse-directed cursor is placed upon the label template 405 shown on the GUI 402 and the computer 408 receives an indication of a button click from the mouse 415 shown in Figure 4. One skilled in the art will instantly recognize that this procedure is a common function-access procedure in the WINDOWS® programming environment. Moreover, one skilled in the art may also recognize numerous other function-access procedures. The application 201 can also receive instructions regarding the label template 405 from other computer programs.

To insert the label template 405 into the program 201, the program 201 must first be running. If the desired printer (e.g., the printer 211) is not set as the default WINDOWS® printer, then the program 201 should select the desired printer. For example, a user may use the familiar WINDOWS® "File," "Print" dialogue associated with the pull-down menus 403 to select the printer 211.

Inserting the label template 405 may be accomplished by pulling down an "Insert" menu in the pull-down menus 403, followed by clicking on the "object" option. As ordinary WINDOWS® users will recognize, this is the typical procedure for inserting an ActiveX object into a program. The label template
5 405 may be one selection choice in a list of possible insertable objects. Accordingly, an ActiveX bar code or serialized text object will be inserted into the application 201 as a label template 405.

The application 201 and the bar-code ActiveX control 202 may provide instructions that customize the label template 405 to specify a desired bar
10 code, provide textual data in association with the bar code, indicate whether the bar-code label should be serialized, as well as specifying other pertinent bar-code properties.

When the label template 405 first appears on the GUI 402, the label template 405 may display a variety of default properties, including a default data
15 set (*e.g.*, "12345678"). The bar-code ActiveX control 202 provides numerous functions for adjusting the label template 405 to suit various needs. The customary "right-click" procedure may be utilized to receive a high-level list of the possible adjustments to the label template 405, such as the preferred bar-code symbology for the label template 405 and whether to serialize the data encoded
20 in the symbol. The application 201 may also be used to adjust the label template 405. The functionality of the application 201 may continue to be accessed following insertion of the label template 405.

The bar-code ActiveX control 202 may further provide selections for the label template 405 associated with ADC industry standards (*e.g.*, the
25 AIAGB3 label standards). The bar-code ActiveX control 202 may also provide multiple label templates 405 in the application 201 and on the GUI 402.

Engaging the label template 405 (through an action such as a "right click") prompts display of a bar-code template properties page such as a template properties page 601 shown in Figure 6, according to an embodiment of the
30 invention. The template properties page 601 provides a facility for altering the

label template 405, including entering specific data to be encoded in the bar-code label. A bar-code selection 602 provides a facility for selecting a bar-code symbology and other bar-code characteristics. A human-readable selection 603 provides a facility for specifying the position and display of the human-readable characters that may accompany a bar code. A font selection 600 provides a facility for specifying the characteristics of any human-readable text that may accompany the bar code. A general selection 605 provides a facility for entering specific data for a bar code and for specifying its orientation. For example, the general selection 605 provides options regarding units of measurement. A serialization selection 606 provides a facility for specifying options associated with serialized text and serialized bar-codes.

Each of the selections 602-606 and the template properties page 601 provides selectable functions that provide additional options and choices. For example, the selections 602-606 may each have associated radio buttons, such as a radio button 607 displayed in association with the serialization selection 606. Actuating the radio button 607 engages the functionality associated with the serialization selection 606. One skilled in the art may readily recognize other controls for engaging the functionality associated with the selections 602-606, such as placing text for the selections 602-606 in a different color and allowing these selections to be made available by placing a cursor over them and clicking a mouse button.

Engaging a radio button 607a associated with the bar-code 602 selection produces a bar-code properties table, such as the bar-code properties table 701 shown in Figure 7, according to an embodiment of the invention. The bar-code properties table 701 provides selections regarding the appearance of a bar-code label. The bar-code properties table 701 includes a symbology-specific options area that provides different choices depending upon which bar-code symbology (bar-code language) has been selected. For example, once Interleaved 2-of-5 has been selected, then a symbol-specific "check digit" option may be displayed, with other non-pertinent options in the bar-code properties

table 701 rendered non-selectable. On the other hand, selecting the Code 128 symbology causes the bar-code ActiveX control 202 to provide symbology-specific choices, such as "UCC/EAN-128" and "code set," both of which are discussed below.

5 A symbology selection 702 provides choices regarding which specific symbology will be utilized for the bar-code template, such as the label template 405. A bar-code symbology provides an encoding scheme that converges character data into a pattern of wide and narrow bars and spaces. According to an embodiment of the invention, the bar code choices provided by
10 the symbology selection 702 include: full ASCII Code 3-of-9, regular Code 3-of-9, Code 93, Code 128, Data Matrix, PDF 417, Codabar, Interleaved 2-of-5, MSI Plessey, UPC-A, UPC-E, EAN-13, EAN-8, 2-digit Supplemental, 5-digit Supplemental, and Postal (Zip, DPBC).

 An x-dimension selection 703 provides choices regarding a basic
15 element of a bar code - the width of its narrowest element. The width of the narrowest element is known as a bar code's "x dimension." Along with a bar code's wide-to-narrow element ratio, the x dimension is a factor in determining a bar code's density. Decreasing a bar code's x-dimension value increases the bar code's density, while increasing the x-dimension value decreases the bar code's
20 density.

 For example, in printing a Code 3-of-9 bar code (with its ratio selection 704 set to "auto") on a 300 dpi Hewlett-Packard LaserJet II printer, the lowest x-dimension value available may be 5.0 mil. With an x-dimension value of 5.0 mil, the density of the bar code will be 12.5 characters-per-inch ("CPI"),
25 the highest possible density setting. On the other hand, the highest x-dimension value available for this example is 168.3 mil, resulting in a density of 0.416 CPI, the lowest possible density setting.

 A ratio selection 704 is a symbology-specific option that is enabled for symbologies such as the full ASCII Code 3-of-9, regular Code 3-of-9,
30 Codabar, Interleaved 2-of-5, and MSI Plessey. The ratio selection 704 refers to

the wide-to-narrow element ratio of a bar code which defines the ratio between the widths of the bar-code's wide and narrow elements. An "auto" default setting for the ratio selection 704 may cause the selection of a "best" ratio (*e.g.*, 2.5 to 1 on some laser printers and 3.0 to 1 on some dot-matrix printers). However, the
5 ratio selection 704 may also allow for the specification of other ratios in small increments, *e.g.*, specifying in one-tenth increments any ratio from 2.0 to 1 through 3.0 to 1.

A density selection 705 provides choices regarding the number of characters that may be encoded per inch in a bar code. The density selection 705
10 may be available for most symbologies, except for the Postal symbology whose standard defines a required density. The selected symbology 702 may affect a bar code's density. Different symbologies require different numbers of bars and spaces to represent the same information. The thickness of the bars and spaces also affects a bar code's density. Narrow bars and spaces require less room to
15 express the bar code. As a bar code's character density increases, the room required to print it decreases, allowing the bar-code message to take up less space. Of course, if the density increases too much, the bar code's readability may decrease. The optics of a bar-code scanning device may not be sufficiently precise to properly detect extremely thin lines and spaces, and the printing
20 tolerance of a printing device may not be sufficiently precise to produce good bar codes at high density settings.

Of course, when printing high-density bar codes on either dot matrix or laser printers, a high-resolution optical scanner may be necessary to read the bar codes. For a variety of reasons (*e.g.*, the age of a dot matrix printer
25 head or toner specks from a laser printer), the narrow white spaces between the bars of a bar code may become too narrow to be detected by some optical scanners. In addition, some laser printers may produce lines that are finer than an optical scanner's optical resolving capability. On the other hand, low-density dot-matrix bar codes may be used to produce high-quality camera-ready artwork.
30 If a low-density bar code is photographically reproduced, the individual dots

from the dot-matrix printer disappear, making the bar code appear to have been printed using a high-quality laser printer.

A height selection 706 provides choices for the heights of the bars in a bar code. The height may be measured from the bar code's top edge to its bottom edge. The height of the bars in the tallest possible bar code can be no longer than the full height of the page. The height of the bars in the shortest possible bar code are typically no shorter than 0.001-inch high, although shorter bars could be made. Generally, very short bar codes should be avoided when using a conventional contact scanner, such as a wand. Bar codes with very short bar heights may be difficult to scan with contact scanners because many users cannot hold their hands sufficiently steady to stay within the bar code's boundaries throughout the scan. This difficulty does not normally occur when using a non-contact scanner, such as a moving beam laser gun. Of course, as scanner resolution continues to improve, these suggestions may no longer be applicable. The height selection 706 may provide default height values for the various bar code symbologies. For example, the Postal bar-code standard has a standard height specification.

As previously discussed, the bar-code properties table 701 includes a number of symbology-specific options, such as a check digit selection 707. The check digit selection 707 is particularly applicable for symbologies such as the full ASCII Code 3-of-9, regular Code 3-of-9, Interleaved 2-of-5 and MSI Plessey. The check digit selection 707 provides choices regarding whether check digits will be added to a bar code. A check digit is an optional character that may be appended to the end of a bar code for error-checking purposes. The check digit's value may be derived from the sum (or weighted sum) of the values of the preceding characters in the bar code. While automated data entry with a bar-code reader may be more reliable than manual data entry with a keyboard (e.g., the keyboard 414 shown in Figure 4), the possibility of decoding errors still exists. For example, a sudden jerk in a scanning motion or a printing defect in a bar-code label may cause a "substitution error" to occur. A substitution error is one

that occurs when one or more characters in a bar-code message transmitted to a computer is different from what is actually encoded into the bar code. In situations where a substitution error might catastrophically affect the integrity of a database, for example, the use of check digits may be highly recommended.

5 A check digit type selection 708 provides choices regarding the type of check digit added to a bar-code label. The check digit type selection 708 may be used with symbologies such as the full ASCII Code 3-of-9 and the regular Code 3-of-9. Check digit type selections may include "MOD 43," which is a check digit type used when printing bar-code labels in symbologies such as
10 Code 3-of-9. The MOD 43 setting allows a check digit to be generated that contains non-numeric (alphabetic) characters in a data string. All alphabetic characters in a bar code may be automatically ignored when another check digit type is selected. The "MSI MOD 10" check digit type may be calculated using the same check digit formula used by MSI Plessey symbology. A "UCC MOD
15 10" check digit may be calculated using the same check digit formula used by symbologies such as Interleaved 2-of-5, UPC, EAN, and UCC/EAN-128. The UCC MOD 10 setting should be used when a bar-code data string contains only numeric data. A "UNISON MOD 10" check digit type provides a check digit that may be calculated using a modified version of the MSI Plessey check digit
20 formula. This check digit may be utilized when a bar-code string contains numeric-only data.

 A UCC/EAN selection 709 provides choices regarding the UCC/EAN bar code, a bar-code standard used worldwide for retail, industrial, and commercial products. The UCC/EAN bar code represents a bar code of the
25 Uniform Code Council ("UCC") and the International Article Numbering Association ("EAN"). The ActiveX control 202 automatically adds certain characters and performs certain functions required in the printing of UCC/EAN-128 bar codes when the UCC/EAN-128 selection 709 has been selected. These functions may include automatically generating a leading FNC1 character,
30 UCC/EAN application identifiers, automatic addition of check digits to certain

fields, and adding extra spaces and parentheses to the human-readable characters.

A code set selection 710 is a symbology-specific option enabled for bar-code symbologies such as Code 128. Although Code 128 can encode all 128 ASCII characters, the Code 128 character set itself contains only 102 characters. Code 128 actually contains three different sets of 102 characters. A start character determines the default character set in use for a Code 128 bar code. However, a Code 128 bar code may switch between different character sets at any point following its start character. Because a single bar code may contain any combination of three character sets, Code 128 offers a high character density when entering bar-code data.

A correction level selection 711 provides correction level choices for use with symbologies such as PDF 417. PDF 417 bar codes may encode such large amounts of information that the possibility of a decoding error occurring during scanning increases relative to the more conventional (1-dimensional) bar codes. Accordingly, simple check digits may not suffice, as they merely detect errors during scanning. Because such errors require another scan, and because PDF 417 is sufficiently complex that the likelihood of additional errors during subsequent scans is significant, the PDF 417 symbology itself employs error correction. PDF 417 encodes enough redundant information into a printed bar-code label that the scanner cannot only detect when errors have occurred, but actually correct them. PDF 417 supports nine levels of error correction, with "0" being the least severe and "8" being the most thorough. When the correction level selection 711 is set to "0," the PDF 417 bar code records very little redundant information, and a scanner may do little more than correct for the simplest errors. When the correction level 711 is set to "8," then significant scanning problems may be corrected. The increased size required for a printed PDF 417 symbol at higher correction settings offsets some of the benefits associated with increased correction. When using PDF 417 in a closed system (e.g., control of both symbol printing and symbol scanning) and with a high degree of confidence in the printing and reading capabilities, the printed size of

the symbols may be successfully reduced by manually specifying a relatively low correction level. On the other hand, if one of these factors is questionable, then a larger printed simple made be desirable.

A truncated selection 712 provides appropriate choices for
5 symbologies such as PDF 417. When available space for printed symbologies
such as PDF 417 is limited, a user may decide to sacrifice some information
typically encoded in the PDF 417 symbol, even though the resulting symbol may
be somewhat harder to read. Engaging the truncated selection 712 substitutes a
single vertical bar for the stop pattern on the right-hand side of the PDF 417
10 symbol and the data column containing the right most four bars in the
checkerboard-looking data area. While choosing the truncated selection 712
saves space, the resulting symbol is typically less robust.

A row height selection 713 provides choices regarding the height of
a row for symbols, such as PDF 417 symbols. The height of each row in a PDF
15 417 symbol is typically set to three times its "x dimension" value (the width of
the narrowest bars in the printed symbol). When a correction level at or above
the standard recommendation has been selected, then the default row height
factor of three is adequate. If a correction level is lower than the standard
recommendation, then the row height factor may be selected above the minimum,
20 for example.

A column selection 714 provides choices for use with symbologies
such as PDF 417. The PDF 417 symbology utilizes multiple rows of
information, with each row resembling a thin bar code. Close examination of
these rows indicates that the first five and the last five bars extend the entire
25 height of the symbol, much like a conventional bar code. These bars are the stop
and start patterns that help a scanner identify the PDF 417 symbology. Inside the
start and stop patterns, the encoded information looks somewhat like an irregular
checkerboard. When the columns selection 714 is set to "auto," the number of
rows and columns in the checkerboard data area attains an overall shape that is as
30 close to a square as possible. The columns selection 714 also allows

specification of the number of columns to appear in the checkerboard data area.

A column consists of four bars and spaces and defines a fundamental unit of encoded data for symbols such as PDF 417. Each row consists of a minimum of three information columns (12 bars and spaces) in the checkerboard portion of the data area. At any given correction level, reducing the number of columns increases the number of rows required to encode the same amount of information. Accordingly, although the number of columns may be reduced in order to make the PDF 417 symbol narrower, doing so will make the symbol proportionately taller. When encoding larger amounts of data, an inappropriate columns setting may result in a PDF 417 symbol that is too tall.

A start character selection 715 provides symbology-specific choices pertinent to symbologies such as Codabar. The start character selection 715 on the bar-code properties table 701 receives instructions regarding which one of the four letters "a," "b," "c," or "d" will be used as the start character for a Codabar bar code. An additional setting of an asterisk "*" identifies that the start character will be supplied in the data. Specifying an "*" when one of the four start characters (a, b, c, or d) is not actually supplied at the beginning of the data for the bar code results in assignment of the "a" start character.

A stop character selection 716 provides symbology-specific choices pertinent to symbologies such as Codabar. The stop character selection 716 on the bar-code properties table 701 receives instructions regarding which one of four letters "a," "b," "c," or "d" will be used as the stop character. In an additional setting, an asterisk "*" identifies that the stop character will be supplied by the data. Specifying an "*" when one of the four stop characters is not actually supplied at the end of the data for the bar code results in assignment of the "a" stop character.

A guard bars selection 717 provides symbology-specific choices for symbologies such as UPC-A, UPC-E, EAN-13 and EAN-8. The UPC/EAN/JAN standards include specifications that some of the bars may extend below the main body of the bar code. These are called "guard" or

“security” bars. For example, in a “standard” full-sized bar code, the main body of the bar code may be 0.900 inches high, with the guard bars extending for another 0.065 inches. If the UPC/EAN/JAN bar codes are of a different height, one may need to adjust the overhang of the guard bars in order to keep them properly proportioned relative to the rest of the bar code and the human-readable text that may accompany the bar code. Accordingly, the guard bars selection 717 provides appropriate choices for making such selections.

A UPC supplemental selection 718 provides choices for symbologies such as UPC-A, UPC-E, EAN-13 and EAN-8. When paperback books and magazines, for example, are bar-coded using the UPC or EAN symbologies, the main UPC/EAN code may be followed by a shorter 2-digit or 5-digit supplemental code immediately to the right of the main code. Characteristic choices (density, height, and position) provided by the supplemental bar-code selection 718 may be based upon the characteristics of the main UPC/EAN code that precedes the supplemental code.

The UPC supplemental selection 718 provides an easy way to add a supplemental bar code to a UPC or EAN bar code. The specifications provided by the supplemental selection 718 may be based on the specifications of the corresponding UPC or EAN bar code. A length option may be used to specify a 2-digit supplemental, a 5-digit supplemental, or even no supplemental bar code, for example. The data for the main UPC bar code and the appended supplemental may be specified as either screen data, prompt data, or input file data. The data for the main UPC bar code and the data for the supplemental bar code may be both specified in the data field for the main UPC bar code. The required number of digits (2 or 5) may be automatically taken from the right side of the data entered in the text selection 604 on the object properties table 601 and provided as a supplemental bar code. The remaining data digits will be printed as a UPC or EAN bar code. If the data does not contain enough digits to print a complete UPC/EAN bar code the data may be padded with leading zeros.

A number system selection 719 provides symbology-specific

choices for symbologies such as UPC-E. UPC-E is an 8-digit symbology in which the first digit is either a "0" or a "1," with the next six digits containing data, and the last digit operating as a check digit. The number system selection 719 specifies whether the first character should be a zero or a one. A user normally supplies the six middle characters of a UPC-E bar code, with a number system digit (the first digit) added to the beginning and a check digit computed and added at the end. However, if the label template 405 receives more than 6 characters, then the bar-code ActiveX control 202 may delete the extra characters. If the label template 405 receives less than 6 characters, then the bar-code ActiveX control 202 may add enough trailing zeros to bring the total number of data characters up to six.

A symbol type selection 720 provides symbology-specific choices for symbologies such as Data Matrix. Like most two-dimensional bar codes, the Data Matrix symbology provides error correction by encoding enough redundant information into a printed symbol that the scanner not only detects when errors have occurred but actually corrects them. The symbol type selection 720 provides options for specifying the level of error correction used in a Data Matrix bar code. The Data Matrix symbology supports 14 levels of error correction, with "ECC 000" being the lowest and "ECC 200" being the highest. When the correction level is set to ECC 000, no redundant information is encoded, and the scanner cannot correct any errors. When set to ECC 200, a large portion of the bar code could be damaged and still be correctly scanned. ECC 200 can significantly increase the size of the bar code. Therefore, depending on the amount of information contained in the bar code, and the amount of room available for printing, sometimes one may choose a lower level of error correction.

A character set selection 721 also provides symbology-specific choices for symbologies such as Data Matrix. The character set selection 721 may be enabled for symbol types other than those like DCC 200. (DCC 200 automatically determines the required character set by evaluating the data

provided.) Depending on the amount of data and type of data contained in a bar code, one may achieve the smallest possible bar code by manually specifying a character set. The choices include: an "auto" option, specifying that the bar-code data will be evaluated to determine the best character set; a "Numeric" option specifying that only numbers will be considered acceptable data; an "Uppercase Alphanumeric with Punctuation" option specifying that only uppercase letters and numbers, including punctuation will be considered acceptable data; an "Uppercase Alphanumeric with Spaces" option specifying that only uppercase letters and numbers, including spaces, will be considered acceptable data; a "Full 128 ASCII" option specifying that all keyboard characters, including upper and lower case letters, numbers and symbols, will be considered acceptable data, and an "All 256 ASCII" option specifying that all 256 ASCII characters, including all keyboard characters and international ASCII characters, will be considered acceptable data. One should specify the character set that most closely matches the data because it will result in the smallest bar code. If the data contains unacceptable characters, the bar-code ActiveX control 202 may display the bar-code label on the user interface 401 with a red "X" in the center and may not allow the bar-code label to be printed.

A use prefix selection 722 provides symbology-specific choices for symbologies such as Data Matrix. The user prefix selection 722 enables selections such as UCC/EAN FNC1 which conform to specific industry standards such as those authorized by AIM international. A "Macro 5" option abbreviates the "05 macro header" (a prefix) and trailer (a suffix) into one symbol character to reduce the number of characters needed to encode data. A "Macro 6" option abbreviates the 06 macro header (a prefix) and a trailer (a suffix) into one symbol character to reduce the number of symbol characters needed to encode data. A "reader initialization" option indicates that the symbol encodes a message used to program the reader system. If a special function character (*e.g.*, FNC1 through FNC6) is specified at the beginning of bar-code data and the use prefix selection 722 has been enabled, the specified prefix will be ignored in favor of the special

function character located in the data. For more information regarding special function characters an interested reader may consult any number of Data Matrix references.

A shape selection 723 provides symbology-specific choices for
5 symbologies such as Data Matrix. The Data Matrix bar code comprises columns and rows wherein data may be stored. Most Data Matrix bar codes contain equal number of columns and rows. However, Data Matrix symbol type ECC 200 allows some bar codes to have more columns than rows. The shape selection 723 provides settings for specifying whether a bar code will have an equal number of
10 columns and rows. A "Square" option draws the bar-code symbol with an equal number of columns and rows. Data Matrix includes 24 square bar-code symbols, ranging from 8 columns and rows to 22 columns and rows. A "rectangle" option draws the bar-code symbol with more columns than rows. Data Matrix includes six rectangle bar-code symbols ranging from 6 rows by 16 columns to 14 rows by
15 22 columns. To use the rectangle selection, the amount of data should be small. If the input data is larger than will fit into allowed rectangle sizes, the bar-code ActiveX control 202 prints the bar-code symbol as a square.

A white on black selection 724 provides symbology-specific choices for symbologies such as Data Matrix. The white on black selection 724
20 allows the bar-code ActiveX control 202 to reverse a Data Matrix bar code for display on both a screen and a printer as white on a black background. When this option is selected, the reverse Data Matrix bar code looks like a white bar code on black box. A different color for a bar code may typically not be selected if the white on black selection 724 has been selected.

25 Engaging a radio button 607b associated with the human-readable selection 603 invokes a human-readable properties table 801 shown in Figure 8, according to an embodiment of the invention. The human-readable properties table 801 provides a facility for adjusting the textual characters that may be displayed with a bar-code label. Human-readable text provides a "backup" for
30 the bar code in the event of error. If the bar code cannot be read, a human

performing/supervising the data-entry task may read the human-readable characters and enter the corresponding data into a computer using an input device such as a keyboard.

Selections in the human-readable properties table 801 provide
5 choices pertaining to the human-readable characters that accompany a bar code on a bar-code label, including whether the human-readable characters should be printed at all. Like the bar-code properties table 701, the human-readable properties table 801 contains a dynamic area that offers different selections depending on which bar-code symbology has been selected.

10 A visibility selection 802 provides choices regarding whether a bar code's corresponding human-readable characters will be included on the bar-code label. The visibility selection 802 may provide optional settings such as "full" and "none." A "full" setting specifies that the human-readable characters will be displayed with the bar code. A "none" setting specifies that the human-
15 readable characters will not be displayed with the bar code.

While bar codes are frequently printed with a corresponding display of human-readable characters, on some occasions users find it advantageous not to display the human-readable characters for security reasons. Not displaying human-readable characters provides a slight measure of added
20 system security. For example, assume an employee ID card is used for tracking time and attendance. Without the display of the human-readable characters on the ID card, a typical person will have difficulty manually entering an employee's identification information into a facility's computer (and perhaps tamper with the information contained therein).

25 A placement selection 803 provides choices regarding whether human-readable characters will be printed above or below an accompanying bar code, such as the bar code 501 shown in Figure 5. The placement selection 803 may have a default setting selection, such as "below the bars." The presence of a default setting frees the user from having to examine every possible choice when
30 creating a bar code.

An alignment selection 804 provides choices regarding alignment characteristics for symbologies such as the full ASCII 3-of-9, the Regular Code 3-of-9, Interleaved 2-of-5, Code 128, and Codabar. The alignment selection 804 provides alignment selections for aligning the human-readable characters that
5 accompany a bar code. A "left" selection causes the human-readable characters to be printed next to one another, with the left edge of the first character in the text stream lining up with the left edge of the first bar in the bar code. The "right" selection causes the human-readable characters to be printed next to one another, with the right edge of the last character in the text string aligned with the
10 right edge of last bar in the bar code. A "centered" selection allows the human-readable characters to be printed next to one another and centered between the edges of the bar code. A "bar density" option allows the human-readable characters to be spaced according to the density of the bar code. In this option, the left edge of each human-readable character may align with the left edge of the
15 beginning of each bar code in the bar-code message.

A vertical offset selection 805 provides choices regarding the vertical distance between the human-readable characters and the bar code when the human-readable characters are displayed either above or below the bar code.

A horizontal offset selection 806 provides choices for adjusting in
20 the position of the human-readable characters in relationship to a bar code. The horizontal offset selection 806 may be used in conjunction with the alignment selection 804. For example, instead of positioning the left edge of the first character in the text string directly in line with the left edge of the first bar in the bar code, the text string may be positioned slightly to the left or to the right. In a
25 sample embodiment, the settings for the horizontal offset selection 806 may default to 0.00 inches (indicating no adjustment), with entry of a positive value moving the characters to the right and entry of a negative value moving the characters to the left.

A character template selection 807 provides choices regarding extra
30 characters and their position in the data string. Bar-code systems and their users

frequently require that extra characters be printed in a bar code's human-readable text that are not part of the scanable bar code. For example, the Code 128 bar codes used on the UCC-128 shipping container labels required by many retail stores need extra spaces and parentheses in the human-readable text. These
5 characters are not part of the bar code and are not read by a bar-code reader scanning the bar code. A dollar sign and decimal point in a bar code represent exemplary "extra" data that may appear in the human-readable characters but not in the scanable bar code. For example, a bar code may contain the numeric string "1099" with an accompanying character string of "\$10.99." In this example the
10 "\$" sign and the "." represent extra data that appear in the human-readable characters but not in the bar code.

A hide check digit selection 808 provides choices for symbologies such as full ASCII 3-of-9, regular Code 3-of-9, Interleaved 2-of-5, UCC-A, UCC-E, EAN-8, and EAN-13. The hide check digit selection 808 choices relate
15 to the display of a check digit in the human-readable characters of a bar code. A default setting may hide the check digit so that the human-readable text will not normally include a check digit character. Activating (or de-activating) a check box associated with the hide check digit selection 808 may cause display of the check digit in the human-readable characters, according to an embodiment of the
20 invention.

A split UCC digits selection 809 provides symbology-specific choices for symbologies such as UCC-A. Symbologies such as UCC-A are 12-digit bar codes. The first digit is the "number system digit," the last digit is the "check digit," and the 10 digits in between contain the data. Conventionally, the
25 first and last digits are separated (or "split") from the center 10 digits. However, a bar-code producer may choose to have a continuous 12-digit human-readable code. The split UCC digits selection 809 may default to a setting that causes the first and last digits to be automatically split from the center 10 digits, but allow the human-readable characters to be toggled to provide a continuous
30 presentation.

A shrink UCC split digits selection 810 provides symbology-specific choices that may be enabled for symbologies such as UCC-A and UCC-E. When the first and last digits of a UCC bar code have been split from the 10 centered digits, the shrink UCC split digits selection 810 directs that the printing of the outer two digits be smaller than the bar code's other human-readable characters. The shrink UCC split digits 810 may be enabled by default so that the split digits will be printed smaller than the other human-readable characters in a UCC bar code unless the setting is changed.

A Code 3-of-9 translation selection 811 provides symbology-specific choices that may be enabled for symbologies such as the full ASCII Code 3-of-9. The Code 3-of-9 translation selection 811 pertains to whether the two-character representation of an ASCII character in a full ASCII Code 3-of-9 bar code will be displayed with the human-readable characters. The character sets of the regular Code 3-of-9 symbology and the full ASCII Code 3-of-9 symbologies both contain 43 bar-code characters. However, in order to represent all 128 ASCII characters, the full ASCII Code 3-of-9 symbology uses special two-character combinations of the 43-character set. Unless specified otherwise, when the human-readable characters of the full ASCII Code 3-of-9 bar codes are printed, the two-character option is automatically converted into the single ASCII character that it represents.

For example, in order to use the full ASCII Code 3-of-9 symbology to bar code the question mark character ("?"), the two-character combination of "%J" is used. The default setting for the Code 3-of-9 translation selection 811 may cause the two-character combination to be automatically converted into the "?" when the human-readable characters are printed. However, changing the default setting for the Code 3-of-9 translation selection 811 to de-select the option, causes both the percent sign "%" and the capital letter "J" to appear in the bar code as human-readable characters.

A show start/stop characters selection 812 provides symbology-specific choices for symbologies such as full ASCII Code 3-of-9, regular Code 3-

of-9, and Codabar. The show start/stop characters selection 812 provides specifications regarding whether to display the start and stop characters in the human-readable characters for symbologies such as Code 3-of-9 and Codabar. The start and stop characters for Code 3-of-9 are often a character such as an asterisk. Some users like to display the start and stop characters as human-readable characters while others prefer display of only the data portion of the bar-code message. Displaying the asterisk allows for a quick visual determination regarding whether Code 3-of-9 is in use. A user printing both Code 3-of-9 and another symbology might find this feature helpful. A default setting may be set to not print the start and stop characters for both the full ASCII and/or regular Code 3-of-9 bar codes. Of course, the show start/stop characters 812 provides for a selection other than the default settings. For Codabar, when the default setting for this option is not enabled, the start and stop characters normally will not be printed with the other human-readable characters in Codabar bar codes. Accordingly, the show start/stop characters selection 812 also provides for activation of this option.

A UCC/EAN template selection 813 provides symbology-specific choices for symbologies such as Code 128. The UCC/EAN-128 bar codes used on the shipping container labels required by some large retail stores, for example, often require extra spaces and parentheses in the human-readable text. Depending on the requirements of the particular application, these characters may be positioned in a variety of ways. A default setting for the UCC/EAN template selection 813, such as "Parentheses and Spaces," may define a common configuration for these characters. The UCC/EAN template selection 813 may provide settings, such as "Parentheses and Spaces," "Parentheses Only," "Parentheses, Spaces, and Split," and "None." The "None" setting is useful when an application specifies that no additional characters will be added to the human-readable string. The "None" setting may also be useful for adding characters to the human-readable string that are not specified in any of the other settings for this selection. Any additional characters may be added using the

character template selection 807.

Engaging a radio button 607c associated with the font selection 604 produces a font selection table 901 shown in Figure 9, according to an embodiment of the invention. The font selection table 901 provides
5 specifications regarding the format of the human-readable characters associated with a bar code.

A typeface selection 902 provides choices regarding the typeface "family" that will be used for each object having text. A typeface "family" is a related group of typefaces that are usually available in different sizes. The
10 typeface selection 902 includes any TrueType fonts installed in the user's computer. Some of the typefaces supplied with WINDOWS® which are available for use include Ariel, Courier New, Symbol, and Times New Roman. Also, additional fonts may be available, in accordance with the printer selected.

A font style selection 903 provides choices regarding whether a
15 font for the human-readable characters associated with a bar code will be in a regular, italic, bold, or bold italic style.

A point size selection 904 receives choices regarding the point size for the human-readable text associated with a bar code. The point size selection 904 provides a list box with scroll bars that can quickly change to the specified
20 text size. Whole and half point sizes may be specified, for example.

A strikeout selection 905 specifies whether the human-readable characters will be printed having a strike-through line. The strikeout selection 905 allows text display while at the same time indicating that the text may be deleted at some future time.

25 An underlined text selection 906 specifies whether the human-readable text will be underlined.

A white on black selection 907 specifies whether the text for the human-readable characters will be displayed and printed as reversed type (white characters on a black background).

30 A horizontal stretch factor 908 provides choices regarding exactly

how much a text object should be distorted from a default setting. Of course, a user may also stretch a text object containing a single line of data by using a mouse and the text object's handles, as is well-known in GUI programs.

The horizontal stretch factor selection 908 allows a value to be
5 specified for the factor by which the text object will either be compressed or stretched horizontally. Assume a default setting is 1.0. The default setting may be increased or decreased as necessary for an application. A number greater than 1.0 stretches the text while a number less than 1.0 compresses the text. For example, if the horizontal stretch factor selection 908 receives a value of 1.75,
10 the text will be stretched to 175 percent of its normal size (1.75×100). Likewise, a text object that receives a horizontal stretch factor value of 0.45 will be compressed to 45 percent of its original size (0.45×100).

When using a horizontal stretch factor selection 908, a user should realize that TrueType and PostScript fonts may be stretched but that device fonts,
15 such as those built into thermal and thermal-transfer printers, cannot normally be stretched. A horizontal stretch factor value is essentially an approximate value for which WINDOWS® provides the closest matching available stretch value. With paragraph text objects, the on-screen handles do not allow one to stretch the text. Instead, these are used to increase and decrease the width of the paragraph
20 itself. In order to compress or stretch the size of the actual text in a paragraph, the horizontal stretch factor selection 908 must be used.

Engaging a radio button 607d associated with the general properties selection 605 produces a general properties table 1001 shown in Figure 10, according to an embodiment of the invention. The general properties
25 table 1001 provides specifications regarding the data for the label template 405.

A text selection 1002 receives data for the label template 405. An orientation selection 1003 provides choices regarding rotation of the bar code. A default setting may rotate the bar code to the right, for example. A units selection 1004 receives specifications regarding whether options will be
30 measured in inches or centimeters. A printer selection 1005 provides

specifications regarding the name of a printer on which the bar-code label will be printed.

Engaging the radio button 607 associated with the serialization selection 606 produces a serialization property table 1101 shown in Figure 11, according to an embodiment of the invention. The serialization property table 1101 provides choices regarding the serialization of data.

A type selection 1102 specifies whether the serialized data will increment or decrement upon serialization. A "serialize by" selection 1103 provides specifications regarding how much the data should change per page printed.

An "identical copies" selection 1104 provides choices regarding how many identical copies of a document will be printed. The serialization property table 1101 for each label template 405 inserted into a document includes an "identical copies" selection 1104. Since multiple label templates 405 contained in a document are independent of each other, it is possible to set the "identical copies" selection 1104 differently for each label template 405. If this happens, and the document is printed, the number of identical copies that print may equal the highest number of identical copies set throughout all the label templates in the document. For example, if two bar codes have been serialized, one set to three copies and the other set to four copies, then four copies will be printed since this was the highest number of copies specified throughout all of the objects. Accordingly, it may be a good idea to set the "identical copies" selection 1104 in only one bar-code object to avoid potential problems.

Most WINDOWS® applications have a setting for "number of copies" that is typically located on a "Print" dialog of the "File" menu. This setting also affects the "identical copies" selection 1104. When using serialized label templates in a document, the application's "copies" selection will change from identical copies to serialized copies. If a user sets the number of copies in an application's Print dialog, the user will also set how many copies will be serialized. Even though the application's "copies" setting now specifies

serialized copies, a user may still access and set identical copies.

An update serial number after print selection 1105 provides specifications regarding the next serial number value to be printed. The serial number may be stored in a text file. Some applications do not always display the
5 correct serial number after a print job is completed. In this case, a user may need to refresh the label template 405 by double-clicking on it then pressing "OK" to invoke its proper redisplay.

A serial number file selection associated with the update serial number after print selection 1105 provides two settings, "serial number file" and
10 "browse." The "serial number file" option receives the name of a text file in which to store the updated serial number. The "browse" option provides a facility for examining a directory structure for a filename for the "serial number file."

A "notes" selection associated with the update serial number after
15 print selection 1105 may be helpful when serialization is not possible. In some situations, the display in the update serial number after print selection dialog box may be grayed except for a "notes" button, which a user may click on for an explanation. The possible causes for not been able to serialize a label template 405 may include: the selected printer is not capable of serializing; the selected
20 printer is not capable of serializing the selected symbology; serialization may not be possible with a human-readable character template enabled; serialization may not be possible with a selected bar density spacing for the human-readable characters; serialization may not be possible with a UCC/EAN option enabled; serialization may not be possible with a check digit enabled (unless it is hidden);
25 serialization may not be possible with human-readable start and stop characters visible, and serialization may not be possible with the postal symbologies human-readable text visible.

Figure 12 shows a bitmap image 1201 such as that provided to the
bitmaps 207 of the GDI 204, according to an embodiment of the invention. The
30 bitmap image 1201 comprises a matrix 1210 composed of bits, such as a bit

1204, each of which represents information. A dark bit, such as the bit 1204, represents data such as a "one," while a light bit, such as a bit 1205, represents data such as a "zero." Note that the bit 1204 is shown as dark in Figure 12, although this coloration is shown for purposes of illustration only, and if printed might actually represent another color. The bitmap image 1201 may include in the matrix 1210 a code section 1203 representing instructions to the printer driver 209, a bar-code bitmap 1202, and security bits, including enumerated bits 1206 in Figure 12. A bar code typically contains too much information to be adequately represented by a mere bit mapping, such as the bar-code bitmap 1202.

10 Accordingly, the code section 1203 together with the bar-code bitmap 1202 may provide complete instructions for the production of a bar-code label, such as may be utilized by the printer driver 209 to print a bar-code label on the printer 211. The code section 1203 includes the choices that have been made with regard to a bar code label, including its symbology, its human-readable text, and other

15 characteristics, *e.g.*, the choices that may be made using the template properties page 601.

The "pattern" of 0s and 1s in the bar-code bitmap 1202 determines the pattern of white and black dots forming a printed bar-code label. The bar-code bitmap 1202 may serve a role similar to that of the bitmap for a font in that

20 the printer driver 209 may use the bar-code bitmap 1202 to determine the general shape of the bar-code label whose final form is determined by the printer driver's programming and the code section 1203.

The security bits, including the bits 1206 may be enabled or disabled for the bitmap image 1201. The security bits prevent a bar-code label

25 from correctly printing when used with a non-proprietary printer driver or a non-proprietary printer. The security bits provide security for the manufacturer of a particular printer because in the absence of specific instructions to remove the security bits from the bar-code label, the resulting label will contain the security bits, rendering the label difficult (if not impossible) to scan. The printer driver

30 209 may have programming instructing it to filter out the security bits and/or the

code section 1203 may inform the printer driver 209 about the pattern used for the security bits in the bit map image 1201.

Figure 13A illustrates the bitmap image 1201 printed as a bar-code label 1301 on a non-proprietary printer (or with a non-proprietary printer driver) with the security bits enabled. The bar-code label 1301 appears as a grayed image. The security bits form the gray pattern in the bar-code label 1301. A typical bar-code scanner cannot scan the bar-code label 1301 because it will be unable to distinguish between the black and gray areas of the bar code label 1301.

Figure 13B illustrates the bitmap image 1201 printed as a bar-code label 1302 on a proprietary printer with the security bits enabled and then filtered, or as printed on a non-proprietary printer with the security bits disabled. The code section 1203 may provide instructions regarding the security bits to the printer driver 209. The printer driver 209 may also be designed to determine which portions of the bitmap image 1201 are security bits, which portions comprise the code section 1203, and which portions comprise the bar-code bitmap 1202. Accordingly, the printer driver 209 processes the security bits so as to remove them in the printed bar-code label, such as the bar-code label 1302. Note that the bar-code label 1302 also includes a human-readable text "12345," although as previously discussed, a bar-code does not require a human-readable text.

Figure 14 is a flowchart illustrating the operations of the bar-code ActiveX control 202, according to an embodiment of the invention.

The bar-code ActiveX control 202 receives a request to insert a label template (e.g., the label template 405) in an application (e.g., the application 202) (step 1401). The bar-code ActiveX control 202 inserts a label template into the application (step 1403).

If the inserted label template is not complete (step 1405), then the bar-code ActiveX control receives instructions regarding the inserted label template (step 1407). The ActiveX control 202 may receive instructions

pertaining to the label template with regard to the various characteristics of the label template's bar-code (e.g., the bar-code properties table 701 shown in Figure 7). The ActiveX control 202 may also receive instructions pertaining to: human-readable properties of the label template (e.g., the human-readable properties table 801 of Figure 8; font characteristics of the label template (e.g., the font properties table 901 of Figure 9; general properties of the label template (e.g., the general properties table 1001 of Figure 10; or serialization characteristics of the label template (e.g., the serialization properties table 1101 of Figure 11).

If the inserted label template is complete (step 1405), then the ActiveX control 202 determines whether a print command is received (step 1409). If the ActiveX control 202 receives a print command (step 1409), then the ActiveX control sends a bitmap of the inserted in label template to a printer driver (the printer driver 209) via the GDI 204 shown in the Figure 2. If the ActiveX control does not receive a print command (step 1409), then the ActiveX control examines whether the inserted label template is complete (step 1405).

Figure 15 illustrates the application 201 and the ActiveX control 202 operating in combination with an automation program 1501, according to an embodiment of the invention. As previously discussed, the application 201 may be any type of application, such as a word processing program or a database program. Many WINDOWS® programs have made their programming capabilities available to operations by other programs through a facility known as "Automation." Automation allows software programs access to the capabilities of other programs, such as spreadsheet applications, word processors, and other personal productivity software. Thus, the applications expose their services to programs as well as to people, thereby sharing such services. In other words, the applications themselves become "programmable." Providing this programmability is known as "automation."

An application may become programmable (e.g., automation) by exposing its services through ordinary COM interfaces. Applications may also expose their services through "disinterfaces" (from "disparate" and

“interface.”) A dispinterface resembles a conventional program interface, such as having methods that allow clients access to the method using an interface pointer, but also differs in significant ways. In particular, dispinterfaces may be easily invoked from clients written in simple languages such as Visual Basic.

5 For example, Microsoft® Excel may be accessed using a dispinterface. Accordingly, Excel makes its internal services available through its dispinterfaces supported by various COM objects, which provide methods such as “average,” “check spelling,” and many more. Applications built on Excel no longer are restricted to using Excel’s built-in macro language but
10 instead can be written in virtually any programming language. Thus, Excel itself becomes a toolbox for the building of new applications.

A user could produce hundreds of bar-code labels using the application 201 and the bar-code ActiveX control 202, manually entering the data for each new bar-code label. On the other hand, if the application 201 provided
15 its interfaces to other programs, through a programming paradigm such as automation, then an automation application 1501 could likewise produce hundreds of bar-code labels without requiring user intervention. As an ordinarily skilled artisan will recognize, the automation application 1501 could even be quite small and simply process a data file of data to be used in bar-code labels by
20 calling on the appropriate interfaces from the application 201 and the bar-code ActiveX control 202.

Figure 16 illustrates a bar-code ActiveX control 1604 utilized in a client/server environment, according to an embodiment of the invention. An ActiveX architecture lets an ActiveX control, such as the bar-code ActiveX
25 control 202, interact with other programs over a network (such as the Internet). The ActiveX architecture uses Microsoft’s COM and distributed COM (“DCOM”) standards. COM allows applications to communicate locally, and DCOM allows applications to communicate over a network. DCOM stipulates how COM components communicate over WINDOWS-compatible networks.
30 DCOM also permits the distribution of different portions of a single computer

program across two or more networked computers. Using DCOM, a computer program's distribution across a network is not apparent to the user.

ActiveX enables the sending of small programs to a browser, without involving other special software on the desktop. ActiveX also provides a
5 number of elements that allow a developer to add more things to an otherwise static Web page. ActiveX controls serve as interactive objects in a Web page to provide interactive and user-controllable functions. ActiveX documents enable users to view non-HTML documents through a web browser. Active scripting controls the integrated behavior of several ActiveX controls and/or Java applets
10 from the browser. A Java virtual machine enables any ActiveX supported browser, such as Internet Explorer 3.0, to run Java applets and to integrate Java applets with ActiveX controls. An ActiveX server framework provides a number of Web server-based functions such as security, database access, and others.

Embodiments of the invention provide bar-code ActiveX controls
15 and bar-code preparation pages having common interface mechanisms between programming elements. Common interface mechanisms are desirable for programs operating on a single computer and are essential when splitting programming elements across a network.

Embodiments of the invention also support the Common Object
20 Request Broker Architecture ("CORBA"), used by companies such as IBM, Sun, and Apple. CORBA allows objects from one program to communicate with objects in other programs even if the two programs are written in different programming languages and run on different computing platforms. A program requests objects through an object request broker ("ORB") and does not need to
25 understand the structure of the program containing the object.

Using object models, such as CORBA and DCOM, embodiments of the present invention provide a bar-code label production facility having no specific language requirements for its constituent elements. Programmers may use any supported language, such as C, C++, Visual Basic ("VB"), Visual Basic

Script ("VBScript"), Java, or JavaScript to manipulate and interact with bar-code labels.

Figure 16 provides a sample bar-code label production program to a user application operating over the Internet, according to an embodiment of the invention. A Windows Internet client 1601 utilizes a Microsoft Internet Explorer program 1602 to send a request over a Hypertext Transfer Protocol ("HTTP") connection 1607 to an Internet server 1605 in order to access an Internet storage medium 1606. The Internet storage medium 1607 contains a bar-code preparation page 1603 that includes a bar-code ActiveX control 1604. The bar-code ActiveX control 1604 accesses a GDI 1611 associated with a bar-code server computer 1610 over a DCOM connection 1615. The bar-code server computer 1610 includes a printer driver 1612 that receives a bitmap image for a bar-code label from the GDI 1611, interprets the bitmap image, and sends appropriate commands for printing a bar-code label to the printer 211.

The bar-code server computer 1609 may be accessed by more than one Windows Internet client at a time such that bar-code labels from multiple applications may print on the printer 211. This flexibility provides a bar-code label production facility with the ability to distribute bar-code label applications to computers throughout an organization while maintaining centralized administration over the bar-code ActiveX control 1604 and the printing of bar-code labels. The bar-code ActiveX control 1604 operates with both DCOM and CORBA distributed object interfaces, as previously discussed. These interfaces provide programmers with considerable latitude in the deployment of bar-code ActiveX-based solutions.

While the application thus far has discussed only the production of bar-code labels, the invention is equally applicable to other ADC symbols and systems, according to an embodiment of the invention. Figure 17 illustrates a system for producing SmartCards, according to an embodiment of the invention.

An application 1701 includes an ActiveX control 1702 that provides services for producing the programming associated with a SmartCard.

The application 1701 produces a bitmap image 1703 that contains code 1704 for a SmartCard's instruction set and SmartCard data 1704 that comprises the data to be encoded in a SmartCard. The application provides the bitmap image 1703 to the bitmaps 207 of the GDI 204. A SmartCard device driver 1706 may be
5 configured as if it were a printer driver. Accordingly, the GDI 204 may provide the bitmap image 1703 to the SmartCard device driver 1706. The SmartCard device driver 1706 utilizes its internal programming to decode the bitmap image 1703 and provide the data 1704 to a SmartCard encoder 1707 that then produces a SmartCard according to its received instructions.

10 Embodiments of the invention provide a system and method for controlling several ADC device platforms. The invention is particularly applicable to applications operating on personal computers, but is equally suitable for use with workstation-type computers and hand-held computing devices, including computing devices having ADC symbol acquisition devices,
15 such as bar-code readers.

Although specific embodiments of, and examples for, the invention are described herein for illustrative purposes, various equivalent modifications are possible within the scope of the invention, as will be recognized by those skilled in the relevant art. The teachings provided herein of the invention can be
20 applied to other ADC symbol production systems, not necessarily the sample bar-code production system described above. Various ADC symbol systems, and accordingly, various ADC symbol production systems can be employed under the invention. While the invention discusses bar-code symbologies, the invention can similarly read visual indicia of other stored images. The invention may be
25 applicable for data other than ADC symbols, including alphanumeric fonts, Unicode characters, and font glyphs.

The invention is not limited to the WINDOWS operating system and may be used in computers having other operating systems, such as Unix, Linux, OS/2, and the Macintosh operating system ("Mac OS").

The invention is not limited to communications through the WINDOWS GDI but may alternatively communicate through other operating system elements, especially operating system elements that provide graphical display interfaces for applications. The invention may operate with any form of a
5 “page object” that is not understood by a graphics processing capability provided by an operating system.

Moreover, embodiments of the invention may be applied to any computing application in which data passes from a programming element through a surrogate (such as the GDI) to another programming element that may
10 utilize the data.

In the examples provided, the bitmap code section, such as the bitmap code section 1203, has been shown at the “top” of the bitmap image. Of course, the code section could be placed anywhere in the bitmap image. In addition, the bar-code bitmap, such as the bar-code bitmap 1202, could be
15 replaced by a coding sequence that provided still further instructions to a printer driver regarding the characteristics of the bar code. In this embodiment of the invention, the security bits might operate differently, *e.g.*, operating as an encryption mechanism or a password mechanism. One skilled in the art, might even recognize other methods for transmitting the printing instructions for a
20 particular bar-code label (or any other ADC symbol) to the printer driver. One skilled in the art may similarly determine equivalent printer instruction communications that do not require the code section 1203 shown in Figure 12.

Printer drivers associated with the invention, such as the printer driver 209, may operate with any type of printer, including but not limited to
25 thermal printers. The invention is also equally applicable with any application that causes a target device (such as the printer driver 209) to perform a series of tasks. For example, a target device that receives ADC symbol instructions and builds a physical expression of the ADC symbol, such as a dipole device or RF tag.

Embodiments of the invention may further include a help system, including a wizard that directs a user through the steps of producing an ADC symbol and sending it for printing (or another form of physical embodiment) on a printer (or other suitable device).

5 The ADC ActiveX controls, such as the bar-code ActiveX control 202, provided by the invention need not necessarily be written as 32-bit applications, unless necessary for operations in conjunction with a particular operating system, such as WINDOWS 95.

As previously discussed, in one embodiment of the invention, the
10 ADC ActiveX control may be obtained by utilizing an "insert" option followed by selecting an "object" option in the pull-down menu provided. While such a procedure may be followed in association with one sample embodiment of the invention, ADC ActiveX controls may be accessed through a variety of approaches, as an ordinarily skilled artisan will easily recognize. In addition, the
15 ActiveX control may receive selections for the label template through other user interfaces than the sample user interfaces shown herein. For example, the display of ADC symbols on a user interface, such as the GUI 402 may utilize the "what-you-see-is-what-you-get" ("WYSIWYG") display paradigm in one embodiment of the invention, while in another embodiment of the invention, the
20 display may be based upon another display paradigm. One skilled in the art may easily think of numerous alternate approaches to providing a user interface to receive the information needed to produce an ADC symbol.

Communications between elements of the invention (*e.g.*, communications between the printer driver 209 and the printer 211) may be
25 conducted over both a wireless communications system and a wired communications system.

According to one embodiment of the invention, Java applets may provide a plug-in ADC symbol producing system for use with an application on both a single computer and in a networked embodiment. Java is an object-
30 oriented programming language similar to C++. Java was designed to be secure

and platform neutral, meaning that Java code may run on any computing platform. Java is a useful language for programming applications for the World Wide Web since users access the web from many different types of computers. Java is especially well adapted for use in programming small applications, or
5 applets, for use in the World Wide Web. A Java applet may be loaded and run by an already running Java application, such as a web browser. Java applets may be downloaded and run by any web browser capable of interpreting Java, such as Microsoft Internet Explorer, Netscape Navigator, and Hot Java.

Under embodiments of the invention, a remote computing system
10 having browsing software may receive an ADC symbol producing plug-in computing applications via an Hypertext Markup Language ("HTML") documents, Dynamic Hypertext Mark-Up Language ("DHTML") documents, Extensible Mark-Up Language ("XML") documents, and/or other documents containing ADC symbol producing plug-in applications over the World Wide
15 Web. A remote computing system, such as the Windows Internet client 1601, uses the ADC symbol producing plug-in application to produce ADC symbols. The remote computing system may further utilize protocols such as SNMP, TCP, UDP/IP, and UDP+ in conjunction with a plug-in ADC symbol producing application. UDP+ provides additional functionality that is not provided by
20 UDP, such as guaranteed packet delivery and packet ordering. UDP+ is further described in U.S. Application No. 08/852,002, entitled, "Reliable Communication over an Unreliable Transport Layer in a Hand-Held Device using User-Configurable Timers," and is also described in U.S. Application No. 08/851,848, "Providing Reliable Communication over an Unreliable Transport
25 Layer in a Hand-Held Device using a Persistent Session," both filed on May 6, 1997 and assigned to a common assignee.

The ADC symbol production system may operate with protocols and languages in addition to those specifically disclosed herein. For example, the ADC symbol production system is not limited to operations using HTTP, HTML,
30 DHTML, XML, UDP, TCP/IP, FTP, SNMP, and TFTP but is equally applicable

to other similar languages and protocols. Similarly, the ADC symbol production system may be developed according to an object-oriented programming methodology or using any other programming methodology that results in a computing system having equivalent functionality.

5 The invention finds equal applicability in stationary data collection terminals, such as a permanently mounted device, and in desktop personal computers. The invention has been discussed in terms of computer programs but is equally applicable for systems utilizing hardware that performs similar functions, such as application specific integrated circuits ("ASICs").

10 These and other changes can be made to the invention in light of the above detailed description. In general, in the following claims, the terms used should not be construed to limit the invention to the specific embodiments disclosed in the specification and the claims, but should be construed to include all ADC symbol production systems, ADC symbol printing systems, and ADC
15 symbol development programs that operate under the claims set forth hereinbelow. Accordingly, the invention is not limited by the disclosure, but instead its scope is to be determined entirely by the following claims.

CLAIMS

What is claimed is:

- 1 1. A method for producing an automatic data collection
2 ("ADC") symbol, comprising:
3 preparing an instruction data set for an ADC symbol in a container
4 application having an inserted ADC preparation application;
5 translating the instruction data set for the ADC symbol into a
6 bitmap image; and
7 sending the bitmap image to an operating system component that
8 receives bitmap images, wherein upon receiving the bitmap image the operating
9 system component identifies a pre-selected printer driver to receive the bitmap
10 image.
- 1 2. The method of claim 1, further comprising:
2 sending the bitmap image to the pre-selected printer driver; and
3 translating the bitmap image into the instruction data set for the
4 ADC symbol by the pre-selected printer driver.
- 1 3. The method of claim 1, further comprising providing
2 security data to the bitmap image such that the ADC symbol cannot be properly
3 produced without removal of the security data.
- 1 4. The method of claim 1, further comprising providing
2 security data to the bitmap image such that the ADC symbol cannot be properly
3 produced without removal of the security data, wherein the pre-selected printer
4 driver has been configured to remove the security data from the data set.
- 1 5. The method of claim 1, further comprising providing ADC
2 symbol preparation instructions by the inserted ADC preparation application that
3 provide at least one of ADC symbology selection, human-readable text selection,
4 ADC data selection, and serialization selection and storing the ADC symbol
5 production instructions in the instruction data set.

1 6. The method of claim 1 wherein the ADC symbol has a
2 selected ADC symbology, further comprising providing ADC symbol production
3 instructions for the ADC symbol including at least one of special characteristic
4 selections for the selected ADC symbology, dimensional specifications for the
5 ADC symbol, and error correction specifications for the ADC symbol and storing
6 the ADC symbol production instructions in the instruction data set.

1 7. The method of claim 1 wherein the ADC symbol has an
2 associated human-readable text, further comprising providing ADC symbol
3 production instructions for the ADC symbol by the inserted ADC preparation
4 application including at least one of a location for the human-readable text,
5 alignment of the human-readable text in association with the ADC symbol,
6 including of ADC check digits in the human-readable text, the inclusion of ADC
7 start and stop characters in the human-readable text, and a font for the human-
8 readable text.

1 8. The method of claim 1, further comprising providing ADC
2 symbol production instructions for the ADC symbol including providing
3 instructions for receiving at least one of data to be expressed by the ADC
4 symbol, an orientation for the ADC data symbol, units of measurement expressed
5 by the ADC symbol, and a production device used to produce the ADC symbol.

1 9. The method of claim 1 wherein the ADC symbol that is one
2 of a bar-code label, a Radio Frequency ("RF") tags, a resonator, a SmartCard, a
3 magnetic strip, an optical character recognition ("OCR") symbol, a recorded
4 speech symbol, a two-dimensional ("2D") symbol, or a dipole device.

1 10. The method of claim 1 wherein the inserted ADC
2 preparation application is one of an ActiveX control or a Java applet.

1 11. The method of claim 1 wherein the operating system is one

2 of a WINDOWS operating system, a Macintosh operating system, a Linux
3 operating system, or a Unix operating system.

1 12. A method for producing an automatic data collection
2 ("ADC") symbol, comprising:

3 preparing an ADC symbol in a first data format;

4 translating the ADC symbol in the first data format into a second
5 data format; and

6 sending the ADC symbol in the second data format to an operating
7 system component that receives data in the second format, wherein after
8 receiving the ADC symbol in the second data format the operating system
9 component identifies a pre-selected driver to receive the ADC symbol in the
10 second data format.

1 13. The method of claim 12, further comprising:

2 translating the ADC symbol in the second data format into an ADC
3 instruction set by the pre-selected driver.

1 14. The method of claim 12 wherein the second data format is a
2 bitmap image.

1 15. The method of claim 12, further comprising providing
2 security data to the ADC symbol in the second data format such that the ADC
3 symbol cannot be properly produced without removal of the security data,
4 wherein the pre-selected driver has been configured to remove the security data
5 from the ADC symbol in the second data format.

1 16. The method of claim 12, further comprising providing ADC
2 symbol preparation instructions that provide at least one of ADC symbology
3 selection, human-readable text selection, ADC data selection, and serialization
4 selection and storing the ADC symbol production instructions in the first data
5 format.

1 17. The method of claim 12 wherein the ADC symbol has a

2 selected ADC symbology, further comprising providing ADC symbol production
3 instructions for the ADC symbol including at least one of special characteristic
4 selections for the selected ADC symbology, dimensional specifications for the
5 ADC symbol, and error correction specifications for the ADC symbol and storing
6 the ADC symbol production instructions in the first data format.

1 18. The method of claim 12 wherein the ADC symbol has an
2 associated human-readable text, further comprising providing ADC symbol
3 production instructions for the ADC symbol including at least one of a location
4 for the human-readable text, alignment of the human-readable text in association
5 with the ADC symbol, including of ADC check digits in the human-readable
6 text, the inclusion of ADC start and stop characters in the human-readable text,
7 and a font for the human-readable text.

1 19. The method of claim 12, further comprising providing ADC
2 symbol production instructions for the ADC symbol including providing
3 instructions for receiving at least one of data to be expressed by the ADC
4 symbol, an orientation for the ADC data symbol, units of measurement expressed
5 by the ADC symbol, and a production device used to produce the ADC symbol.

1 20. The method of claim 12, further comprising providing ADC
2 symbol production instructions for the ADC symbol including providing
3 instructions for receiving at least one of a serialization control for the ADC
4 symbol and an update control for the ADC symbol following production of the
5 ADC symbol on a production device.

1 21. The method of claim 12 wherein the ADC symbol that is
2 one of a bar-code label, a Radio Frequency ("RF") tags, a resonator, a
3 SmartCard, a magnetic strip, an optical character recognition ("OCR") symbol, a
4 recorded speech symbol, a two-dimensional ("2D") symbol, or a dipole device.

1 22. The method of claim 12 wherein the operating system

2 component has not been programmed to process data in the first data format.

1 23. The method of claim 12 wherein the operating system
2 component is an element of one of a WINDOWS operating system, a Macintosh
3 operating system, a Linux operating system, or a Unix operating system.

1 24. A method for processing data, comprising:
2 preparing a data set of a first type to resemble a data set of a second
3 type;
4 providing the prepared data set to an operating system element,
5 wherein the operating system element processes data of the second type but not
6 of the first type;
7 identifying a programming element for receipt of the prepared data
8 set by the operating system element; and
9 providing the prepared data set to the identified programming
10 element, wherein the identified programming element processes data of the
11 second type.

1 25. The method of claim 24 wherein preparing the data set of
2 the first type further comprises securing the prepared data set for use only with a
3 set of programming elements.

1 26. The method of claim 24 wherein preparing the data set of
2 the first type further comprises securing the prepared data set for use only with a
3 set of programming elements, wherein the identified programming element is in
4 the set of programming elements.

1 27. The method of claim 24 wherein the data set of the first type
2 comprises instructions for preparing an automatic data collection ("ADC")
3 symbol.

1 28. The method of claim 24 wherein the data set of the second
2 type comprises a bitmap.

1 29. The method of claim 24 wherein the data set of the first type
2 comprises instructions for preparing an automatic data collection ("ADC")

3 symbol and the data set of the second type comprises a bitmap.

1 30. The method of claim 24 wherein the operating system
2 element is the graphical display interface ("GDI") of the WINDOWS operating
3 systems.

1 31. The method of claim 24 wherein the identified programming
2 element is a printer driver.

1 32. In a computing system that provides automatic data
2 collection ("ADC") symbol preparation, a computer-readable medium having a
3 data structure, comprising:

4 an instruction field that contains ADC symbol preparation
5 instructions for an ADC symbol; and

6 a bitmap of the ADC symbol.

1 33. The computer-readable medium of claim 32, further
2 comprising:

3 a security code that prevents the ADC symbol from being produced
4 properly on unauthorized production machines.

1 34. The computer-readable medium of claim 32 wherein the
2 ADC symbol that is one of a bar-code label, a Radio Frequency ("RF") tags, a
3 resonator, a SmartCard, a magnetic strip, an optical character recognition
4 ("OCR") symbol, a recorded speech symbol, a two-dimensional ("2D") symbol,
5 or a dipole device.

1 35. A system for producing bar-code labels, comprising:

2 a general-purpose computing application;

3 a programmed bar-code preparation application that has been
4 inserted into the general-purpose computing application;

5 an operating system having a generalized graphics display
6 component; and

7 a printer driver that receives bar-code label instructions from the
8 programmed bar-code preparation application via the generalized graphics
9 display component.

1 36. The system of claim 35 wherein the programmed bar-code
2 preparation application sends a bitmap to the generalized graphics display
3 component that sends the bitmap to the printer driver, wherein the bitmap
4 includes the bar-code label instructions.

1 37. The system of claim 35 wherein the programmed bar-code
2 preparation application is an ActiveX control.

1 38. The system of claim 35 wherein the operating system is the
2 WINDOWS operating system and the generalized graphics display component is
3 the graphics display interface ("GDI").

1 39. A system for producing an automatic data collection
2 ("ADC") symbol, comprising:

3 a container application in executable form;

4 an ADC preparation application that provides ADC symbol
5 production instructions and has been configured for insertion into the container
6 application;

7 a display interface in an operating system that receives drawing
8 data from the ADC preparation application; and

9 an output driver that receives the drawing data from the display
10 interface, wherein the drawing data contains ADC symbol production
11 instructions.

1 40. The system of claim 39 wherein the display interface has not
2 been programmed to process displays pertaining to ADC symbols.

1 41. The system of claim 39 wherein the ADC preparation
2 application is one of an ActiveX control or a Java applet.

1 42. The system of claim 39 wherein the operating system is one
2 of a WINDOWS operating system, a Macintosh operating system, a Linux
3 operating system, or a Unix operating system.

1 43. The system of claim 39 wherein the drawing data is a
2 bitmap and wherein the ADC preparation application has provided the ADC
3 symbol production instructions in the bitmap.

1 44. The system of claim 39 wherein the display interface is the
2 graphics display interface ("GDI") provided in the WINDOWS operating system.

1 45. The system of claim 39 wherein the ADC preparation
2 application provides security data to the ADC symbol production instructions
3 such that an ADC symbol cannot be properly produced without removal of the
4 security data.

1 46. The system of claim 39 wherein the ADC preparation
2 application provides security data to the ADC symbol production instructions and
3 wherein the output driver has been configured to remove the security data from
4 the ADC symbol production instructions.

1 47. The system of claim 39 wherein the ADC preparation
2 application prepares ADC symbol production instructions that provide at least
3 one of ADC symbology selection, human-readable text selection, ADC data
4 selection, and serialization selection.

1 48. The system of claim 39 wherein the ADC preparation
2 application prepares ADC symbol production instructions for an ADC symbol

3 having a selected ADC symbology and at least one of special characteristic
4 selections for the selected ADC symbology, dimensional specifications for the
5 ADC symbol, and error correction specifications for the ADC symbol.

1 49. The system of claim 39 wherein the ADC preparation
2 application prepares ADC symbol production instructions for an ADC symbol
3 having an associated human-readable text and the ADC preparation application
4 has been further configured to provide instructions for producing the human-
5 readable text including at least one of a location for the human-readable text,
6 alignment of the human-readable text in association with the ADC symbol,
7 including of ADC check digits in the human-readable text, the inclusion of ADC
8 start and stop characters in the human-readable text, and a font for the human-
9 readable text.

1 50. The system of claim 39 wherein the ADC preparation
2 application prepares ADC symbol production instructions for an ADC symbol
3 and the ADC preparation application has been further configured to provide
4 instructions for receiving at least one of a data set to be expressed by the ADC
5 symbol, an orientation for the ADC data symbol, units of measurement expressed
6 by the ADC symbol, and a production device used to produce the ADC symbol.

1 51. The system of claim 39 wherein the ADC preparation
2 application prepares ADC symbol production instructions for an ADC symbol
3 and the ADC preparation application has been further configured to provide
4 instructions for receiving at least one of a serialization control for the ADC
5 symbol and an update control for the ADC symbol following production of the
6 ADC symbol on a production device.

1 52. The system of claim 39 wherein the ADC symbol is one of a
2 bar-code label, a Radio Frequency ("RF") tags, a resonator, a SmartCard, a
3 magnetic strip, an optical character recognition ("OCR") symbol, a recorded

4 speech symbol, a two-dimensional ("2D") symbol, or a dipole device.

1 53. A system for producing automatic data collection ("ADC")
2 symbols, comprising:

3 a storage unit having an ADC preparation program that prepares
4 ADC symbols;

5 a web browser configured to access the storage unit over a network
6 and retrieve the ADC preparation program from the storage unit;

7 a display component in an operating system that receives a data set
8 in a display format from the web browser; and

9 a driver component that receives the data set from the display
10 component and translates the data set into ADC symbol instructions.

1 54. The system of claim 53 wherein the display component has
2 not been programmed to process displays pertaining to ADC symbols.

1 55. The system of claim 53 wherein the ADC preparation
2 program is one of an ActiveX control or a Java applet.

1 56. The system of claim 53 wherein the display format is a
2 bitmap format and wherein the ADC preparation program has provided the ADC
3 symbol production instructions in the data set in the bitmap format.

1 57. The system of claim 53 wherein the display component is
2 the graphics display interface ("GDI") provided in the WINDOWS operating
3 system.

1 58. The system of claim 53 wherein the ADC preparation
2 program provides security data to the data set such that an ADC symbol cannot
3 be properly produced from the data set without removal of the security data.

1 59. The system of claim 53 wherein the ADC preparation
2 program prepares ADC symbol production instructions that provide at least one
3 of ADC symbology selection, human-readable text selection, ADC data
4 selection, and serialization selection and stores the ADC symbol production
5 instructions in the data set.

1 60. The system of claim 53 wherein the ADC preparation
2 program prepares ADC symbol production instructions for an ADC symbol
3 having an associated human-readable text and the ADC preparation program has
4 been further configured to provide instructions for producing the human-readable
5 text including at least one of a location for the human-readable text, alignment of
6 the human-readable text in association with the ADC symbol, including of ADC
7 check digits in the human-readable text, the inclusion of ADC start and stop
8 characters in the human-readable text, and a font for the human-readable text.

1 61. The system of claim 53 wherein the ADC preparation
2 program prepares ADC symbol production instructions for an ADC symbol and
3 the ADC preparation program has been further configured to provide instructions
4 for receiving at least one of data to be expressed by the ADC symbol, an
5 orientation for the ADC data symbol, units of measurement expressed by the
6 ADC symbol, and a production device used to produce the ADC symbol.

1 62. The system of claim 53 wherein the data set expresses an
2 ADC symbol that is one of a bar-code label, a Radio Frequency ("RF") tags, a
3 resonator, a SmartCard, a magnetic strip, an optical character recognition
4 ("OCR") symbol, a recorded speech symbol, a two-dimensional ("2D") symbol,
5 or a dipole device.

1 63. An automatic data collection ("ADC") preparation
2 application, comprising:
3 an ADC symbol preparer that has been configured to produce ADC
4 symbol instructions;
5 an ADC preparation interface that has been configured to receive
6 ADC symbol characteristic selections and provide received ADC symbol
7 characteristics selections to the ADC symbol preparer; and
8 an ADC instruction transmitter that has been configured to transmit
9 ADC symbol instructions to a display interface in an operating system that
10 receives drawing data.

1 64. The ADC preparation application of claim 63, further
2 comprising:
3 a container interface that has been configured to provide
4 instructions received from a container application to the ADC preparation
5 interface.

1 65. The ADC preparation application of claim 63 wherein the
2 ADC preparation interface has been further configured to provide ADC symbol
3 characteristic selections.

1 66. The ADC preparation application of claim 63 wherein the
2 ADC symbol preparer has been further configured to access a default setting for
3 an ADC symbol characteristic of the ADC symbol characteristics if a value for
4 the ADC symbol characteristic has not been provided by the ADC preparation
5 interface.

1 67. An automatic data collection ("ADC") production driver
2 that sends ADC symbol production instructions to an ADC production device,
3 comprising:

4 an ADC instruction receiver that has been configured to receive
5 ADC instructions from an operating system component in a first format; and

6 an ADC instruction processor that has been configured to access
7 the ADC instructions in the first format and use the ADC instructions to send
8 commands to the ADC production device to produce an ADC symbol.

1 68. The ADC production driver of claim 67 wherein the first
2 format has not been designed as an ADC symbol format and wherein the ADC
3 instruction processor has been further configured to translate the first format into
4 a second format suitable in producing commands to the ADC production device.

1 69. The ADC production driver of claim 67 wherein the first
2 format includes a security format that obscures the ADC instructions and the
3 ADC instruction processor has been further configured to remove the security
4 format to access the ADC instructions.

1 70. The ADC production driver of claim 67 wherein the
2 production driver has been configured to produce ADC symbols in at least one of
3 a bar-code format, a SmartCard format, an RF tag format, an optical character
4 recognition ("OCR") format, a recorded speech format, or a dipole device format.

1 71. A computer-readable medium whose contents cause a
2 computer system to produce an automatic data collection ("ADC") symbol, by
3 performing the steps of:

4 preparing an instruction data set for an ADC symbol in a container
5 application having an inserted ADC preparation application;

6 translating the instruction data set for the ADC symbol into a

7 bitmap image; and
8 sending the bitmap image to an operating system component that
9 receives bitmap images, wherein upon receiving the bitmap image the operating
10 system component identifies a pre-selected printer driver to receive the bitmap
11 image.

1 72. The computer-readable medium of claim 71, further
2 comprising:
3 sending the bitmap image to the pre-selected printer driver; and
4 translating the bitmap image into the instruction data set for the
5 ADC symbol by the pre-selected printer driver.

1 73. A computer-readable medium whose contents cause a
2 computer system to produce an automatic data collection ("ADC") symbol, by
3 performing the steps of:
4 preparing an ADC symbol in a first data format;
5 translating the ADC symbol in the first data format into a second
6 data format; and
7 sending the ADC symbol in the second data format to an operating
8 system component that receives data in the second format, wherein after
9 receiving the ADC symbol in the second data format the operating system
10 component identifies a pre-selected driver to receive the ADC symbol in the
11 second data format.

1 74. The computer-readable medium of claim 73, further
2 comprising:
3 translating the ADC symbol in the second data format into an ADC
4 instruction set by the pre-selected driver.

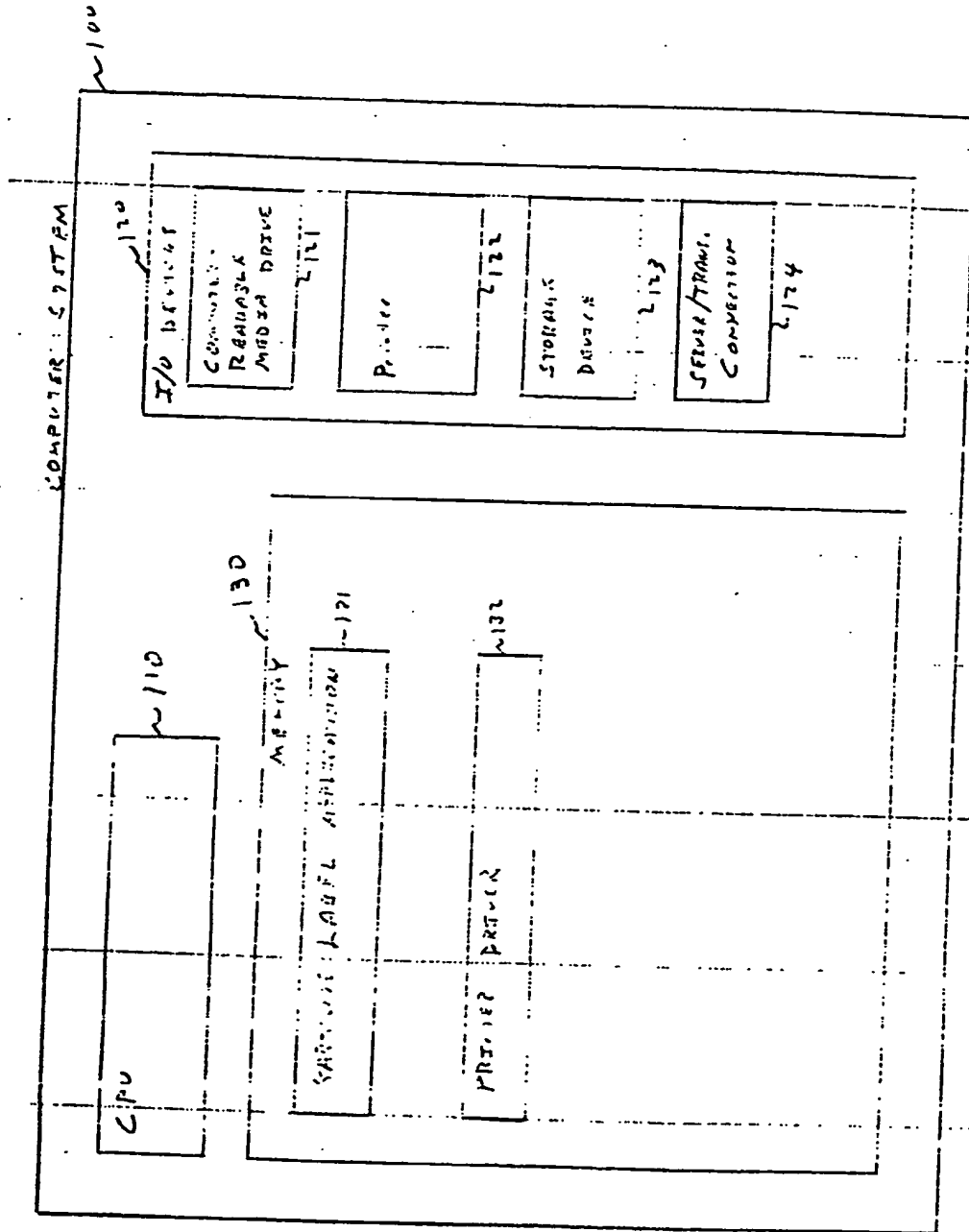


FIG. 1

2/17

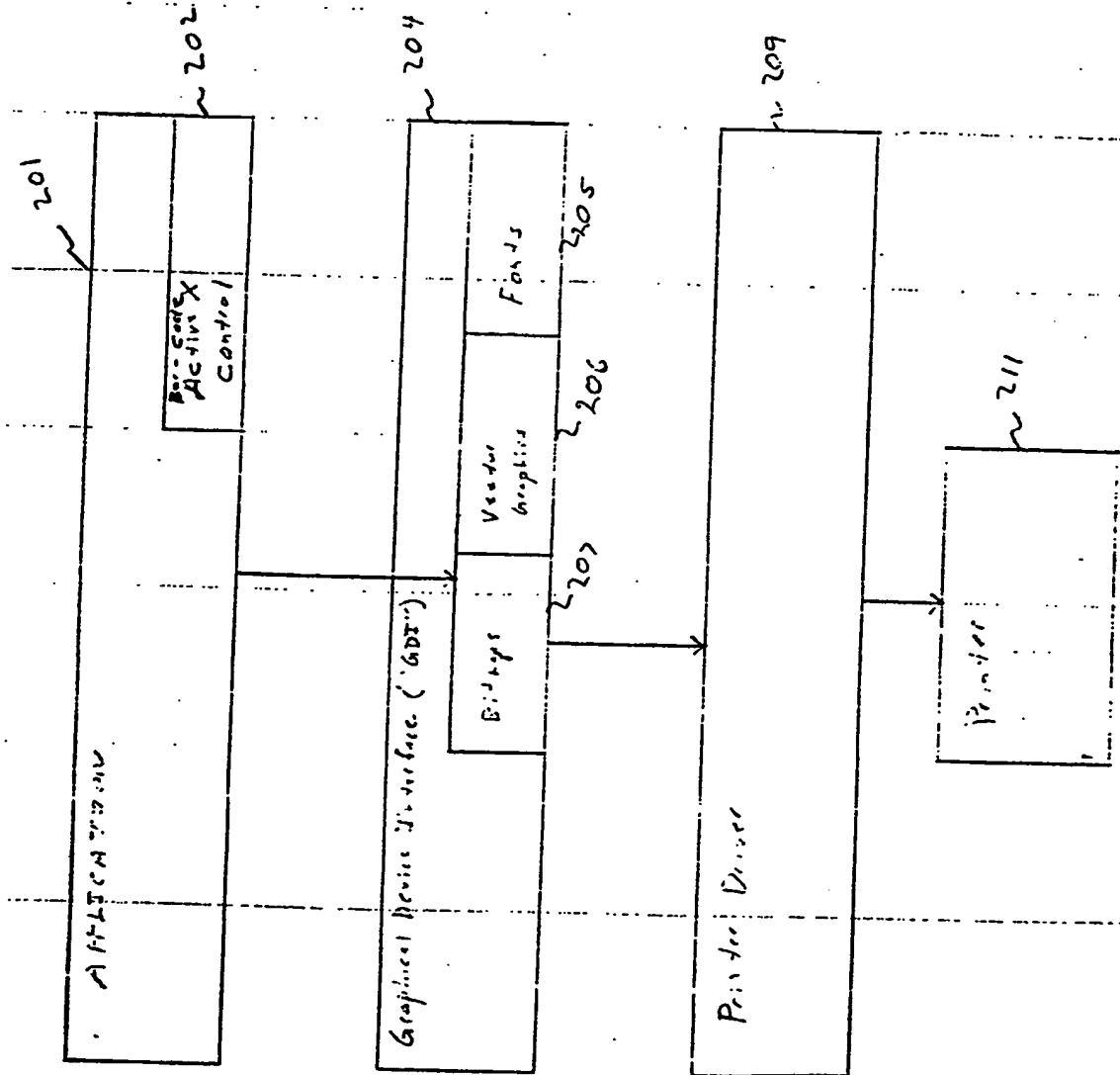
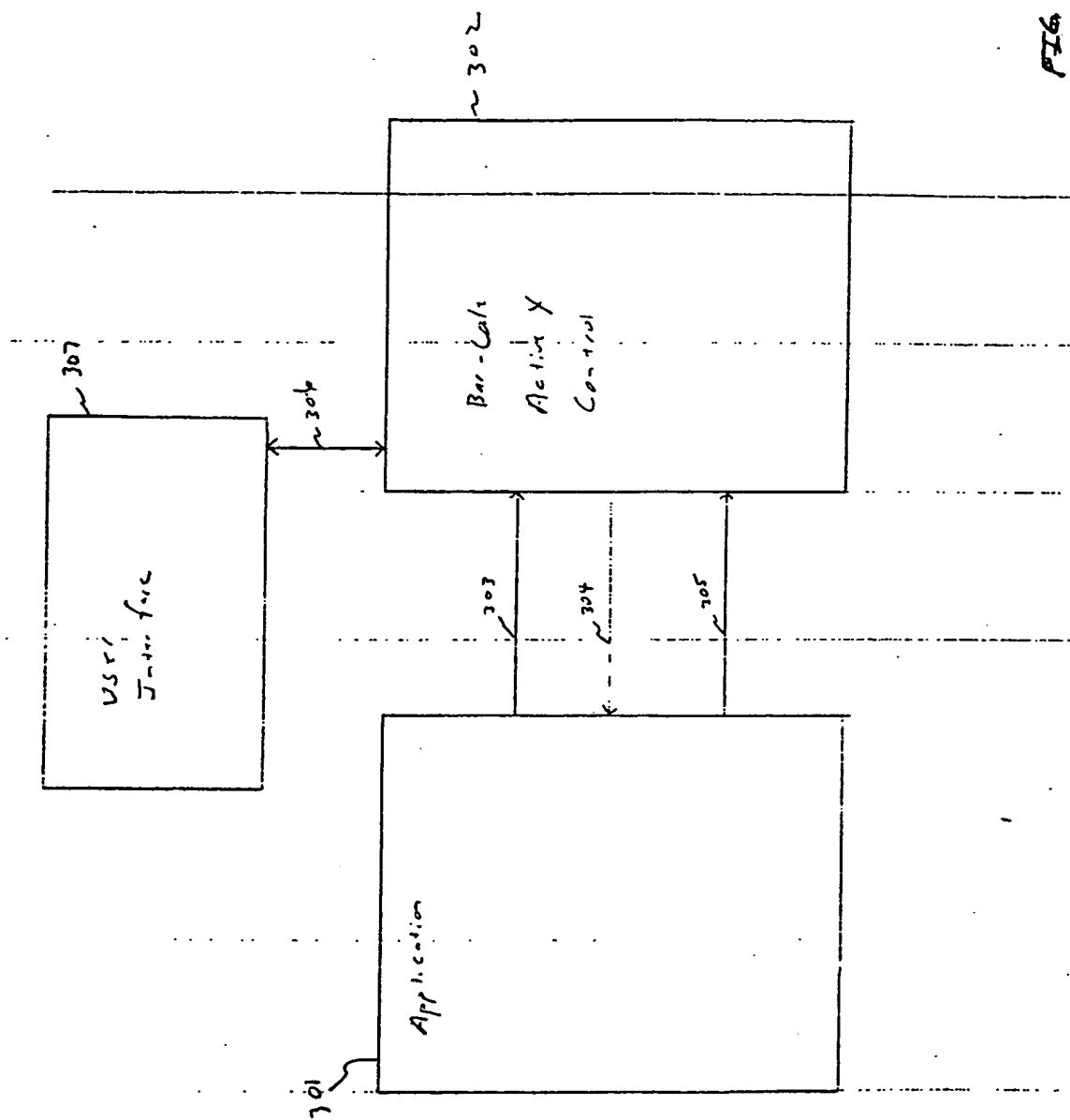


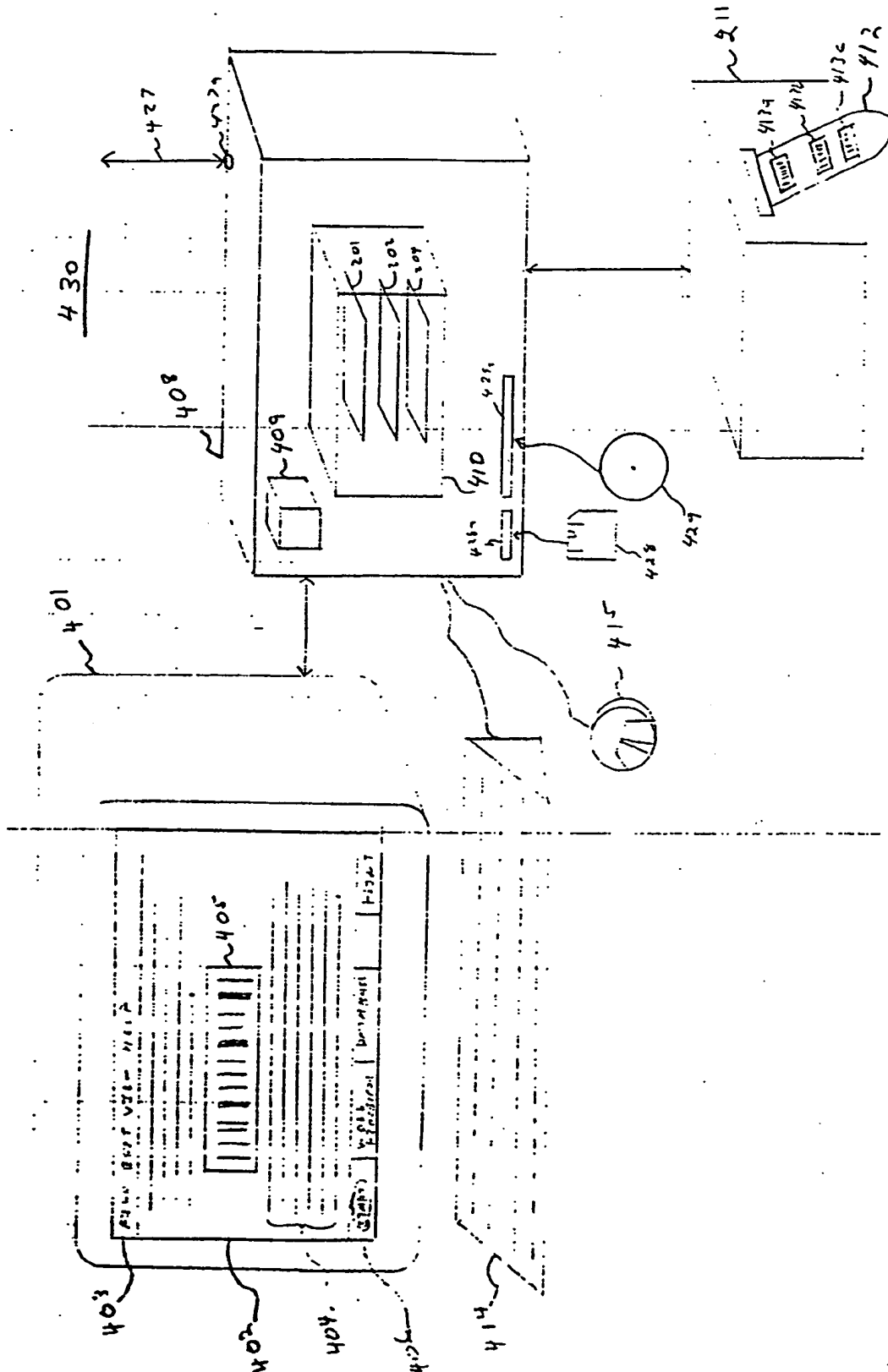
FIG 2

3/17

FIG. 3

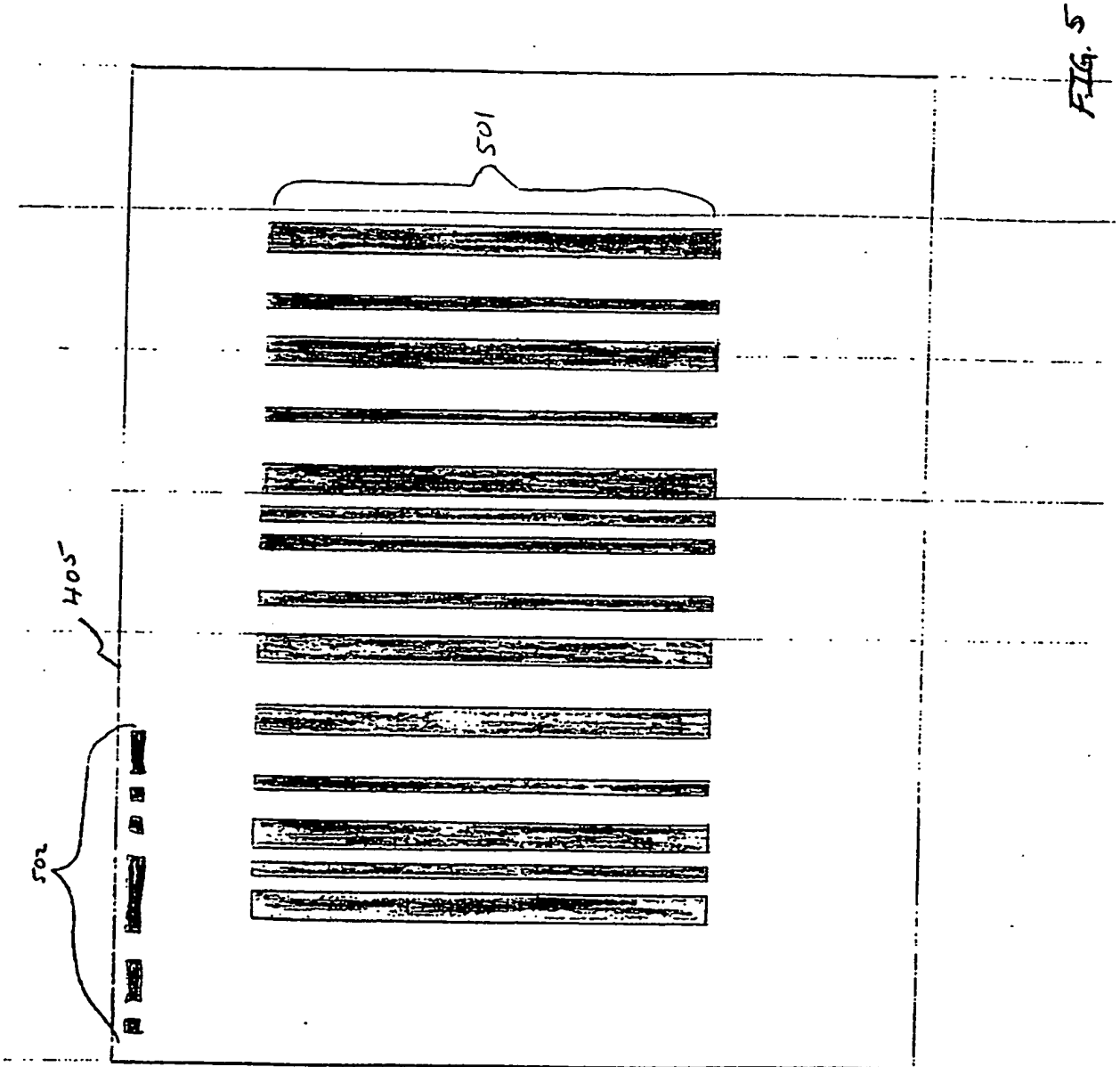


4/17



4514

5/17



6/17

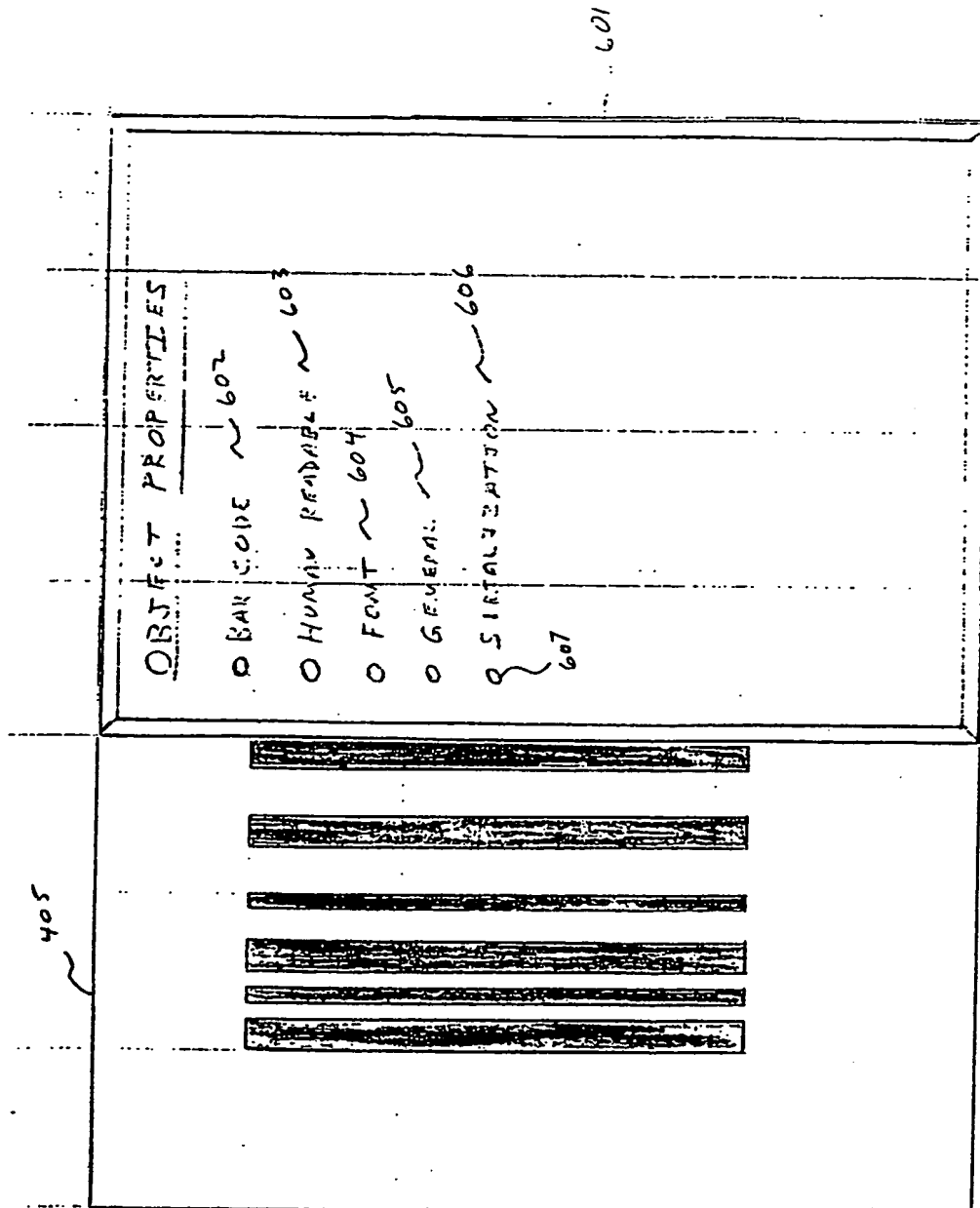


FIG. 6

7/17

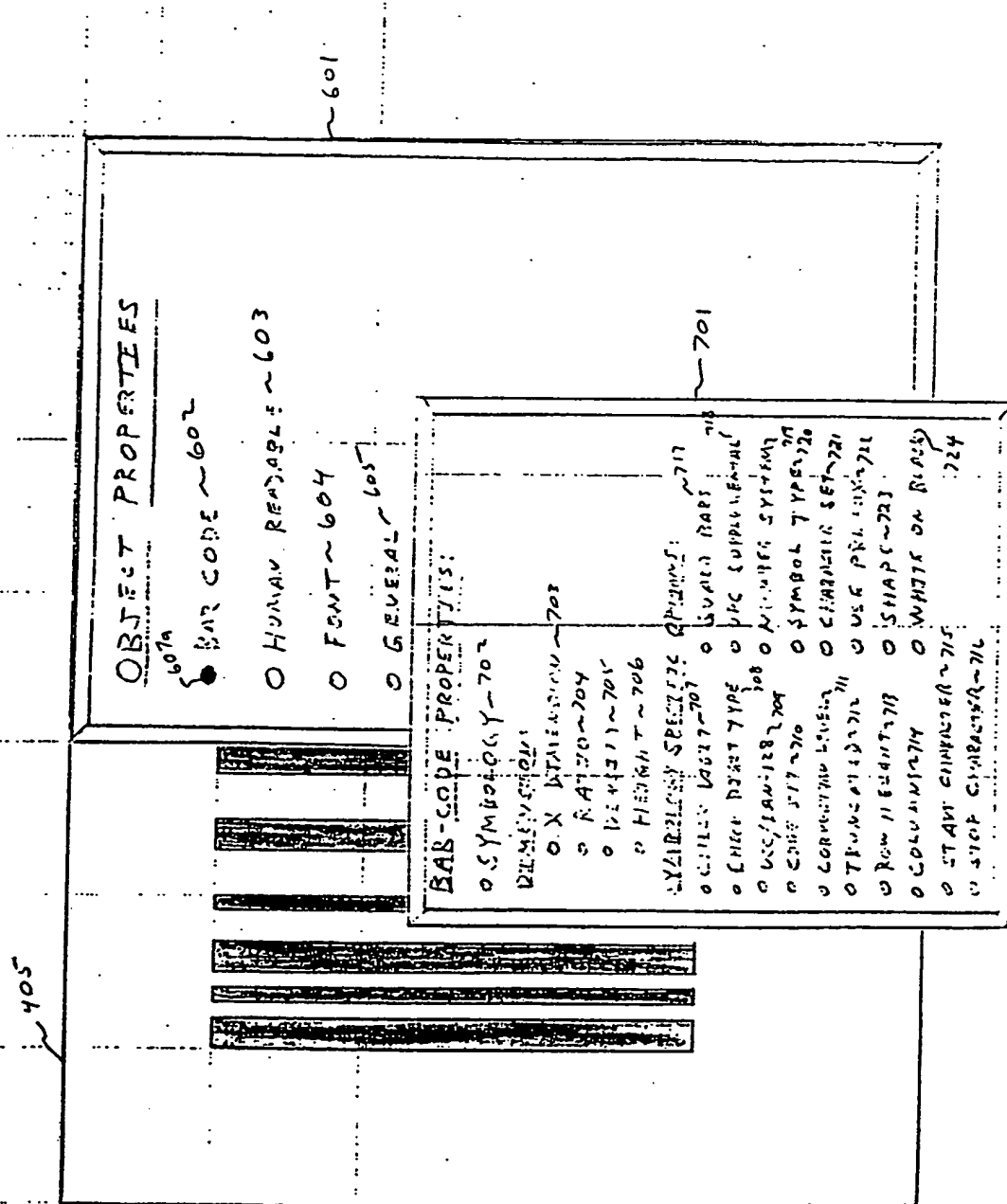
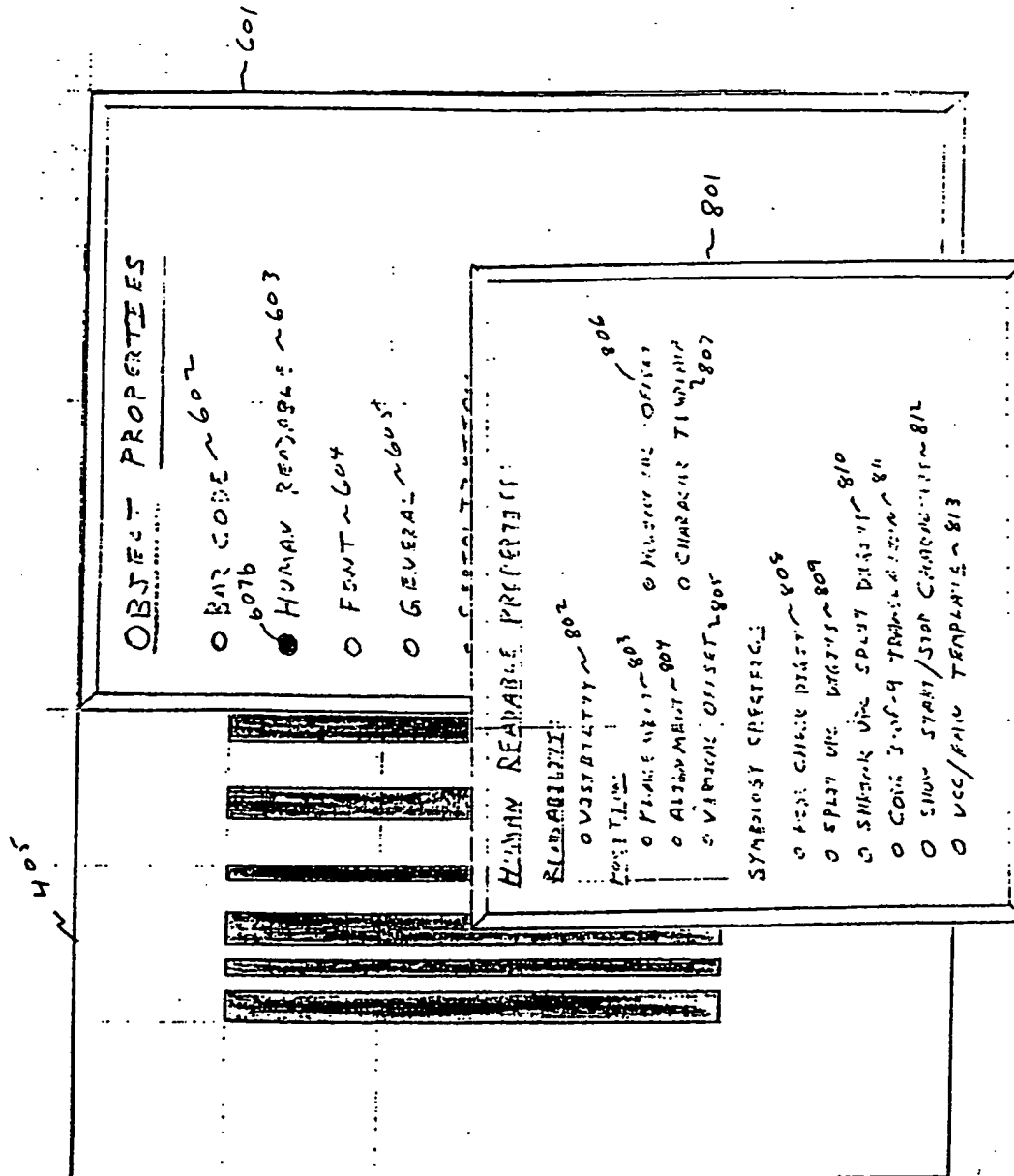


FIG. 7

8/17



8. अथ

8/17

8. अथ

9/17

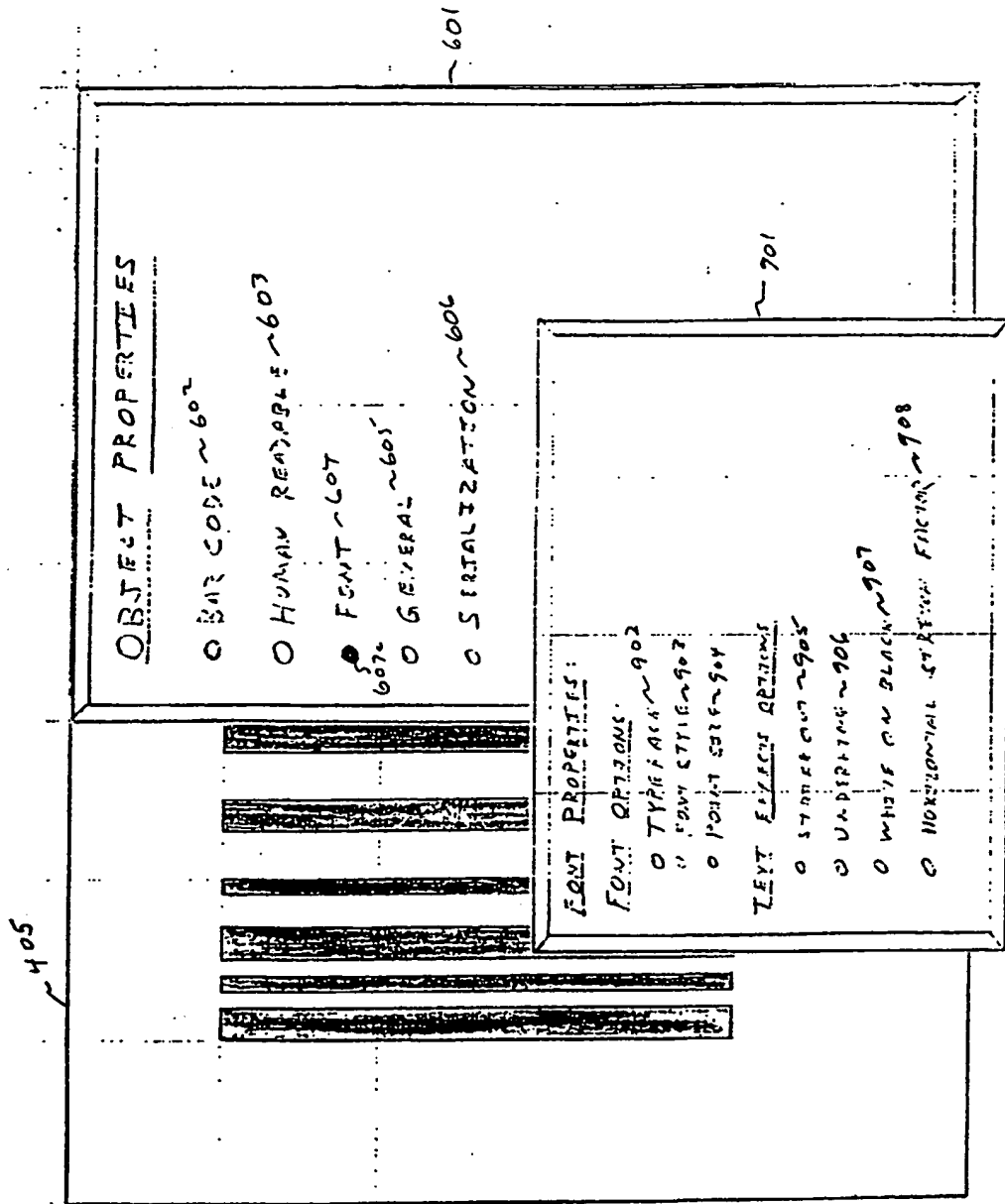


FIG. 9

10/17

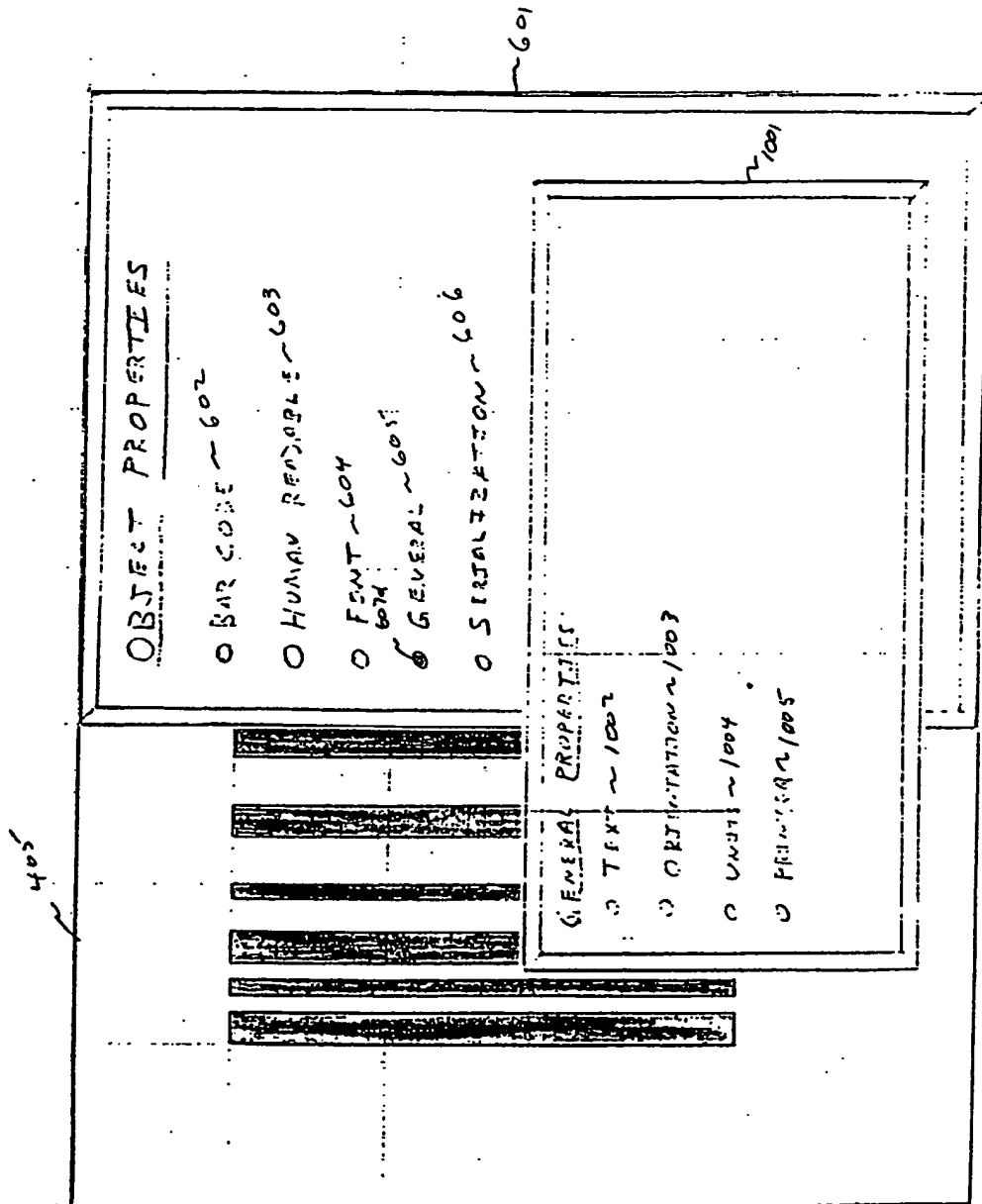


FIG. 10

11/17

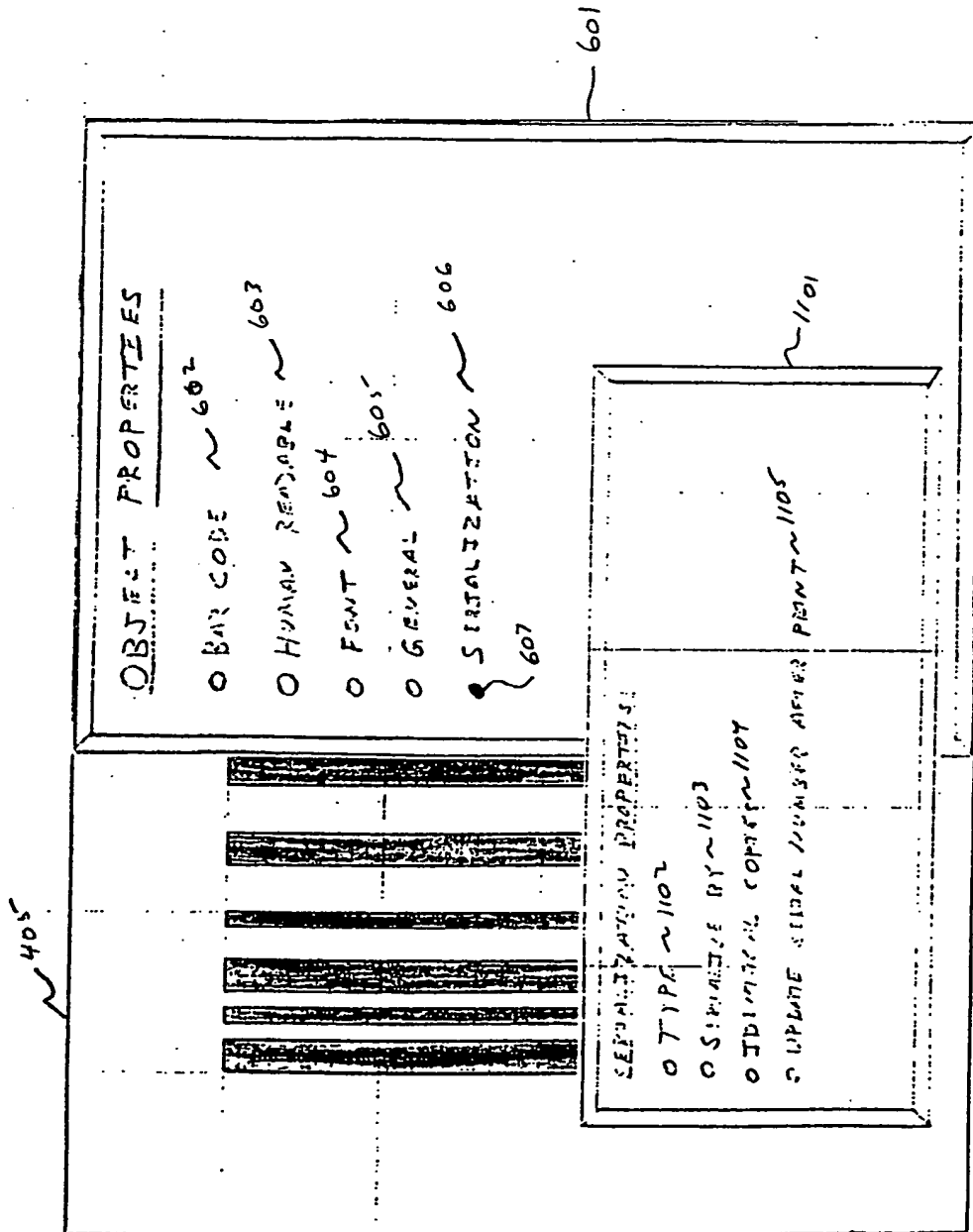


FIG. 11

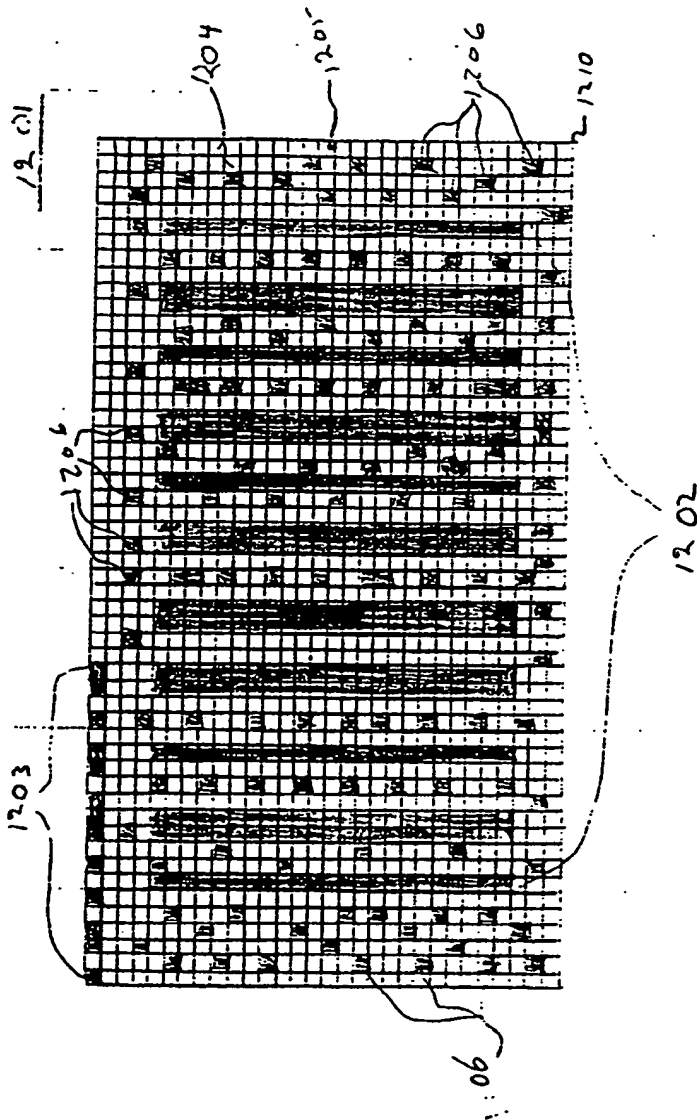


FIG. 12

FIG 13A

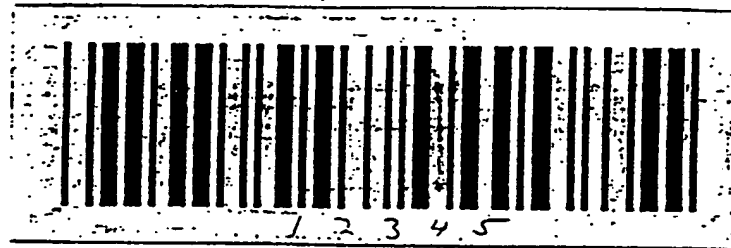
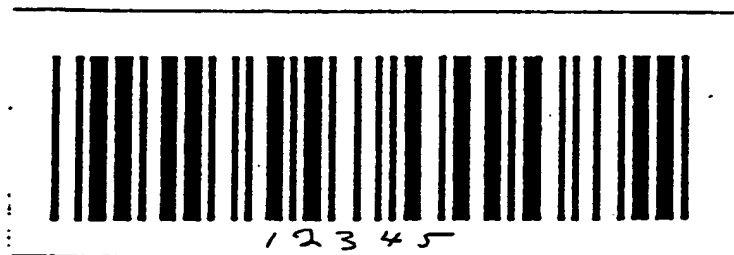


FIG 13B



14/17

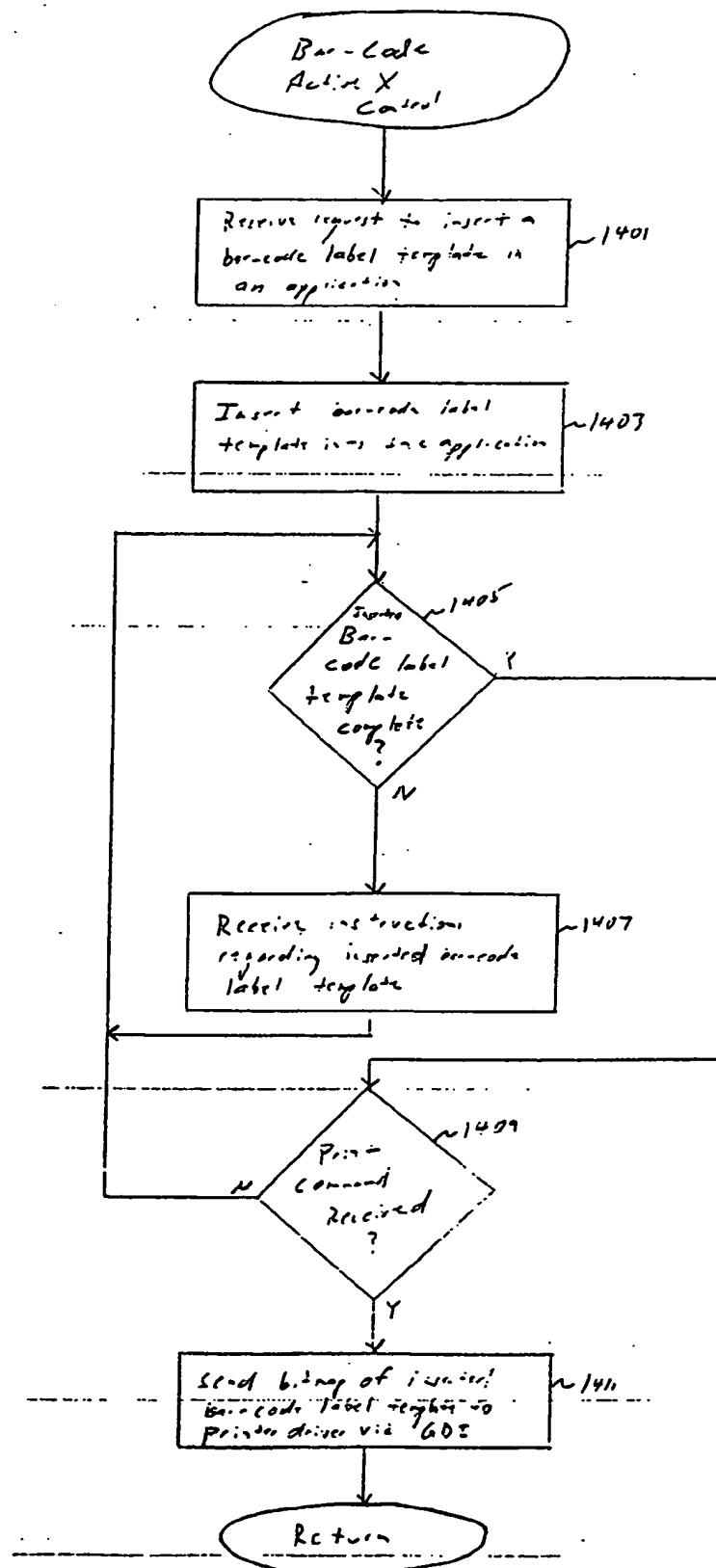
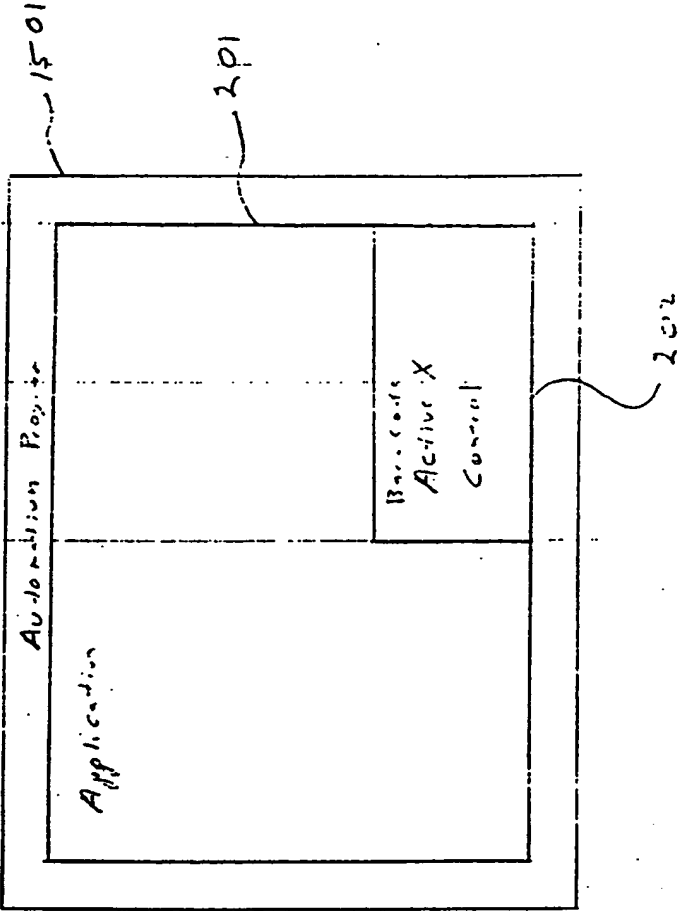


Fig. 14

FIG. 15



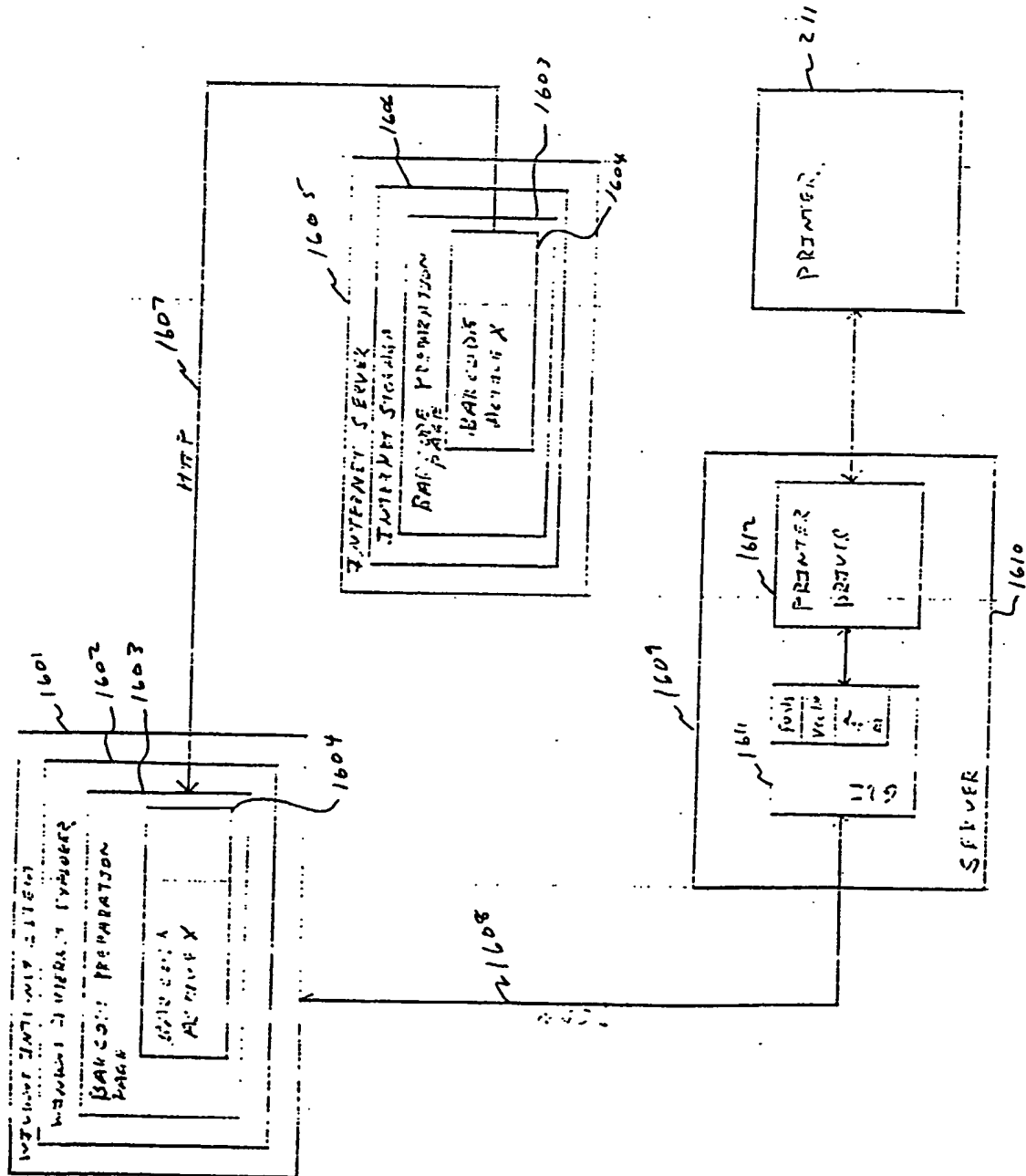


FIG. 16

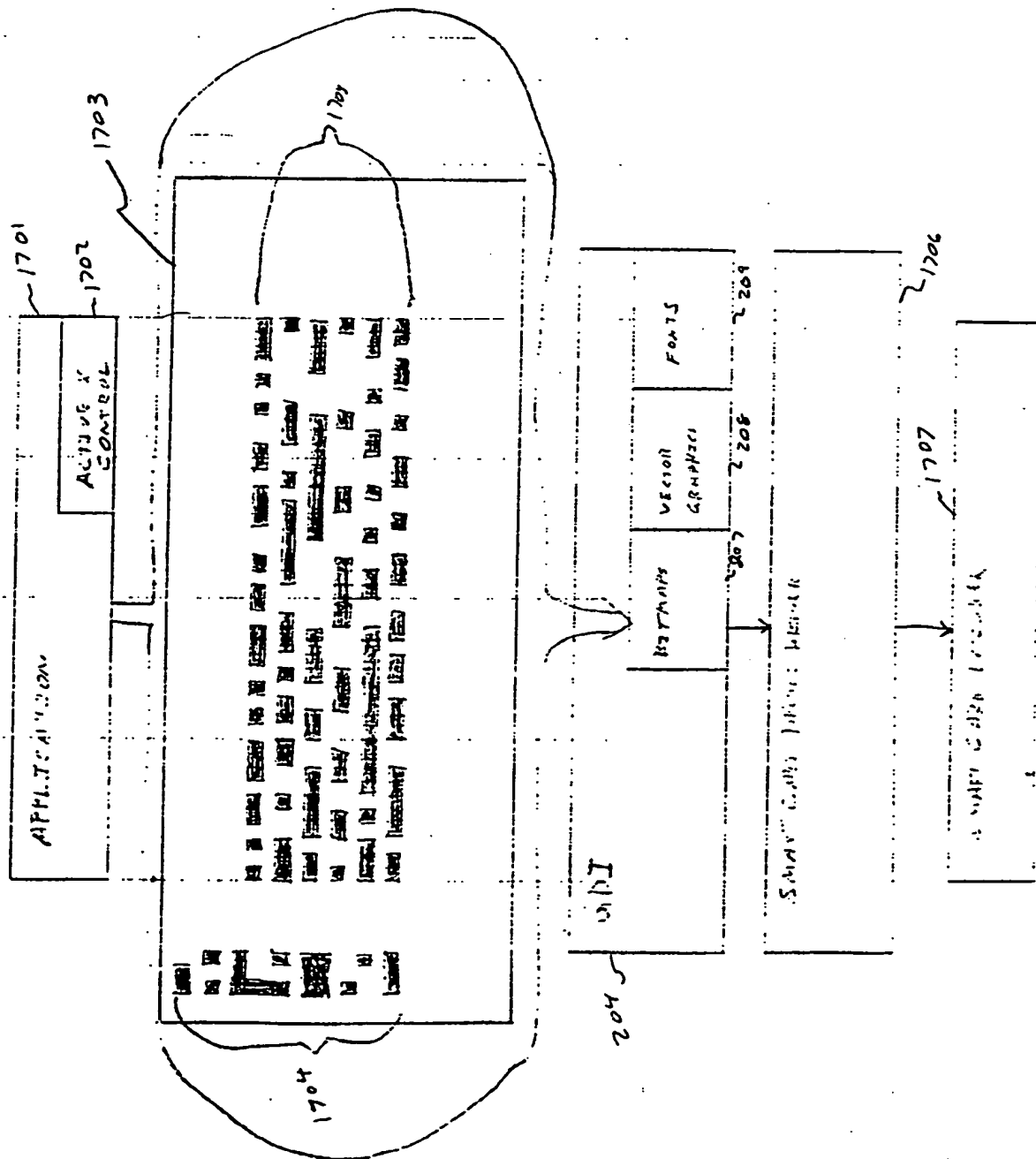


FIG. 17

INTERNATIONAL SEARCH REPORT

International Application No.

PCT/US 00/08405

A. CLASSIFICATION OF SUBJECT MATTER
IPC 7 G06K1/12

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)
IPC 7 G06K G07B

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the International search (name of data base and, where practical, search terms used)

EPO-Internal, WPI Data, PAJ

C. DOCUMENTS CONSIDERED TO BE RELEVANT

| Category * | Citation of document, with indication, where appropriate, of the relevant passages | Relevant to claim No. |
|------------|---|---|
| X | MARZO U: "JavaBarCode Page (3of9-102.zip)" , 'Online! 17 December 1997 (1997-12-17), XP002141521 Retrieved from the Internet: <URL:http://www.geocities.com:0080/Silicon Valley/Pines/4619/3of9-102.zip> 'retrieved on 2000-06-28! | 1,2, 5-14, 18, 19, 21-24, 35, 38-44, 47-50, 52-57, 59-67, 70-74, 32 |
| A | the whole document | |



Further documents are listed in the continuation of box C.



Patent family members are listed in annex.

* Special categories of cited documents:

"A" document defining the general state of the art which is not considered to be of particular relevance

"E" earlier document but published on or after the international filing date

"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)

"O" document referring to an oral disclosure, use, exhibition or other means

"P" document published prior to the international filing date but later than the priority date claimed

"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention

"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone

"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art.

"&" document member of the same patent family

Date of the actual completion of the international search

4 July 2000

Date of mailing of the international search report

17/07/2000

Name and mailing address of the ISA

European Patent Office, P.B. 5818 Patentlaan 2
NL - 2280 HV Rijswijk
Tel. (+31-70) 340-2040, Tx. 31 651 epo nl,
Fax (+31-70) 340-3016

Authorized officer

Schauler, M

INTERNATIONAL SEARCH REPORT

Int. Application No.

PCT/US 00/08405

| C.(Continuation) DOCUMENTS CONSIDERED TO BE RELEVANT | | |
|--|---|--|
| Category * | Citation of document, with indication, where appropriate, of the relevant passages | Relevant to claim No. |
| X | MILGRAM J: "Bookland.py" , 'Online! 7 February 1999 (1999-02-07), XP002141522 Retrieved from the Internet: <URL:http://www.cgpp.com/bookland/bookland .py> 'retrieved on 2000-06-28! | 12,13, 24, 27-29, 31,32, 34,73,74 |
| A | the whole document | 1,35,39, 53,63, 67,71 |
| X | US 5 298 731 A (ETT ALLEN H) 29 March 1994 (1994-03-29) | 24,27-29 |
| A | column 4, line 16-47 | 1,12,32, 35,39, 63,67, 71,73 |
| X | EP 0 824 241 A (CANON APTECS KK) 18 February 1998 (1998-02-18) | 32,34 |
| A | column 4, line 54 -column 5, line 10 | 1,12,24, 35,39, 53,63, 67,71,73 |
| X | TEC IT DATENVERARBEITUNG GMBH: "TBar Code ActiveX (Version 1.4)" , 'Online! 1 January 1999 (1999-01-01), XP002141523 Retrieved from the Internet: <URL:http://www.tec-it.co.at> 'retrieved on 2000-06-29! | 1,2, 7-14,18, 19, 21-23, 35-44, 47-50, 52, 63-68, 70,73,74 |
| | the whole document | |

INTERNATIONAL SEARCH REPORT

Information on patent family members

Inter. Appl. No.

PCT/US 00/08405

| Patent document cited in search report | | Publication date | Patent family member(s) | | Publication date |
|--|---|------------------|-------------------------|------------|------------------|
| US 5298731 | A | 29-03-1994 | NONE | | |
| EP 0824241 | A | 18-02-1998 | JP | 10055401 A | 24-02-1998 |
| | | | JP | 10086492 A | 07-04-1998 |
| | | | US | 6002844 A | 14-12-1999 |

**This Page is Inserted by IFW Indexing and Scanning
Operations and is not part of the Official Record**

BEST AVAILABLE IMAGES

Defective images within this document are accurate representations of the original documents submitted by the applicant.

Defects in the images include but are not limited to the items checked:

- ☐ BLACK BORDERS
- ☐ IMAGE CUT OFF AT TOP, BOTTOM OR SIDES
- ☒ FADED TEXT OR DRAWING
- ☐ BLURRED OR ILLEGIBLE TEXT OR DRAWING
- ☐ SKEWED/SLANTED IMAGES
- ☐ COLOR OR BLACK AND WHITE PHOTOGRAPHS
- ☐ GRAY SCALE DOCUMENTS
- ☐ LINES OR MARKS ON ORIGINAL DOCUMENT
- ☐ REFERENCE(S) OR EXHIBIT(S) SUBMITTED ARE POOR QUALITY
- ☐ OTHER: _____

IMAGES ARE BEST AVAILABLE COPY.

As rescanning these documents will not correct the image problems checked, please do not report these problems to the IFW Image Problem Mailbox.